**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**

**PHẠM VŨ THANH PHONG**

**BÀNH LÊ VŨ THIỆN**

**KHÓA LUẬN TỐT NGHIỆP**

**XÂY DỰNG VÀ PHÁT TRIỂN HỆ THỐNG CHALLENGE GAME TRÊN NỀN TẢNG THIẾT BỊ DI ĐỘNG VỚI SPEED RUNNER**

**KỸ SƯ/ CỬ NHÂN NGÀNH KỸ THUẬT PHẦN MỀM**

**TP. HỒ CHÍ MINH, 2015**

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**

**PHẠM VŨ THANH PHONG – 11520286**

**BÀNH LÊ VŨ THIỆN – 11520378**

**KHÓA LUẬN TỐT NGHIỆP**

**XÂY DỰNG VÀ PHÁT TRIỂN HỆ THỐNG CHALLENGE GAME TRÊN NỀN TẢNG THIẾT BỊ DI ĐỘNG VỚI SPEED RUNNER**

**KỸ SƯ/ CỬ NHÂN NGÀNH KỸ THUẬT PHẦN MỀM**

**GIẢNG VIÊN HƯỚNG DẪN**

**PGS.TS . VŨ THANH NGUYÊN**

**TP. HỒ CHÍ MINH, 2015**

DANH SÁCH HỘI ĐỒNG BẢO VỆ KHÓA LUẬN

Hội đồng chấm khóa luận tốt nghiệp, thành lập theo Quyết định số …………………… ngày ………………….. của Hiệu trưởng Trường Đại học Công nghệ Thông tin.

* 1. …………………………………………. – Chủ tịch.
  2. …………………………………………. – Thư ký.
  3. …………………………………………. – Ủy viên.
  4. …………………………………………. – Ủy viên.

MỤC LỤC

MỤC LỤC iv

DANH MỤC HÌNH viii

DANH MỤC BẢNG x

DANH MỤC TỪ VIẾT TẮT xi

TÓM TẮT KHÓA LUẬN 1

LỜI MỞ ĐẦU 2

CHƯƠNG 1: GIỚI THIỆU VỀ ĐỀ TÀI 3

*1.1* *LÝ DO CHỌN ĐỀ TÀI.* 3

*1.2* *MỤC TIÊU ĐỀ TÀI* 4

*1.3* Ý *NGHĨA ĐỀ TÀI* 4

1.3.1 Đối với người chơi nói riêng và ngành công nghiệp game nói chung 4

1.3.2 Đối với bản thân nhóm 4

1.4 Đối tượng nghiên cứu: 5

1.5 Phạm vi nghiên cứu: 5

1.5.1 Ngôn ngữ C++ và engine Cocos2d-x: 5

1.5.2 Ngôn ngữ Java và Jetty serverlet container: 5

1.5.3 Redis 5

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 6

2.1 Cocos2d-x. 6

2.1.1 Giới thiệu 6

2.1.2 Giấy phép 6

2.1.3 Các tính năng 7

2.1.4 Kiến trúc engine Cocos2d-x 9

2.1.4.1.1 CCScene 10

2.1.2.4.2 CCNode 12

2.1.2.4.3 CCAction 14

2.1.2.4.4 Scheduled 20

2.1.2.4.5 CCDirector 20

2.1.5 Những điểm mạnh của cocos2dx: 21

2.1.6 So sánh với các 2d game engine khác 22

2.2 Jetty 23

2.2.1 Giới thiệu 23

2.2.2 Các tính năng 23

2.2.3 So sánh với các server application khác 24

2.3 Redis 25

2.3.1 Giới thiệu 25

2.3.2 Đặc tính 25

2.3.3 So sánh với các no-sql database khác 25

2.4 Quadtree 26

2.4.1 Định nghĩa 26

2.4.2 ứng dụng quadtree trong game 27

2.5 CÁC KHÁI NIỆM LIÊN QUAN. 27

2.5.1 Tổng quan về Mã nguồn mở. 27

2.5.1.1 Giới thiệu mã nguồn mở. 27

2.5.1.2 So sánh mã nguồn mở và mã nguồn đóng 27

2.5.1.3 Giới thiệu phần mềm mã nguồn mở 28

2.5.1.3.1 Khái niệm 28

2.5.1.3.2 Lợi ích của phần mềm mã nguồn mở 30

2.5.1.3.3 Hạn chế của phần mềm mã nguồn mở 34

2.5.1.3.4 Các loại phần mềm mã nguồn mở 35

2.5.1.4 Giấy phép mã nguồn mở 38

2.5.1.4.1 Định nghĩa 38

2.5.1.4.2 Tính pháp lý của giấy phép mã nguồn mở 39

2.5.1.4.3 Phân loại giấy phép mã nguồn mở 39

2.5.1.5 Các loại giấy phép thông dụng 41

2.5.1.5.1 Giấy phép GNU 41

2.5.1.5.2 Giấy phép BSD 51

2.5.1.5.3 Giấy phép MIT 52

2.5.1.5.4 Giấy phép Apache 54

2.5.1.5.5 Giấy phép Artistic 56

2.5.1.6 Vần đề tương thích của các giấy phép 57

2.5.1.7 Chọn giấy phép phù hợp 61

CHƯƠNG 3: Hiện thực hóa 63

3.1 Client-side 63

3.1.1 Các use case chính 63

3.1.1.1 Mô hình Use-case 63

3.1.1.2 Đặc tả Use-case 64

3.1.2 Các màn hình 67

3.1.2.1 Sơ đồ màn hình 67

3.1.2.2 Mô tả các màn hình 68

3.1.2.2.1 Màn hình chính 68

3.1.2.2.2 Màn hình bảng xếp hạng 69

3.1.2.2.3 Màn hình chọn thông tin map 70

3.1.2.2.4 Màn hình chỉnh sửa map 73

3.1.2.2.5 Màn hình chơi game: 74

3.2 Server-side 78

3.2.1 Kiến trúc 78

3.2.1.1 So sánh các kiến trúc 78

3.2.1.2 Mô hình kiến trúc server 79

3.2.1.2.1 Mô hình giản lược 79

3.2.1.2.2 Mô hình chi tiết 79

3.2.2 Tầng web service 80

3.2.2.1 Cấu hình Jetty 80

3.2.2.2 Cấu hình JSON response 81

3.2.3 Tầng business logic 81

3.2.4 Tầng data access layer 81

3.2.4.1 Hash entity 81

3.2.4.2 List entity 81

3.2.5 Tầng database 83

3.2.5.1 Cấu hình redis 83

3.2.5.2 Danh sách các “bảng” 83

CHƯƠNG 4: THỬ NGHIỆM 84

4.1 QUY TRÌNH THỬ NGHIỆM. 84

4.1.1 Các thành phần của hệ thống thử nghiệm. 84

4.1.2 Tiến hành thử nghiệm. 84

4.2 MỘT VÀI HÌNH ẢNH KẾT QUẢ THỬ NGHIỆM. 84

4.2.1 Màn hình chơi game 84

4.2.1.1 Không sử dụng quadtree 85

4.2.1.2 Sử dụng quadtree 85

4.2.2 Màn hình chỉnh sửa map 86

4.2.3 Màn hình bảng xếp hạng map 87

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 88

5.1 KẾT QUẢ ĐẠT ĐƯỢC. 88

5.1.1 Đối với đề tài 88

5.1.2 Đối với các thành viên trong nhóm 88

5.2 Hạn chế 88

5.3 HƯỚNG PHÁT TRIỂN 88

5.3.1 Về mặt chức năng 88

5.3.2 Về mặt kỹ thuật 89

5.4 Những khó khăn và cách giải quyết 89

5.5 Những bài học thu được 89

TÀI LIỆU THAM KHẢO 90

DANH MỤC HÌNH

[Hình 2.1 Kiến trúc Cocos2d-x 9](#_Toc439087865)

[Hình 2.2 Khám phá các lớp của Cocos2d-x 10](#_Toc439087866)

[Hình 2.3 Chụp màn hình của một game Cocos2d-x 10](#_Toc439087867)

[Hình 2.4 Sơ đồ cây của một màn 11](#_Toc439087868)

[Hình 2.5 Vị trí của một nút con khi được thêm vào nút cha 12](#_Toc439087869)

[Hình 2.6 Sơ đồ các lớp kế thừa từ CCNode 13](#_Toc439087870)

[Hình 2.7 Sơ đồ lớp kế thừa từ CCAction 15](#_Toc439087871)

[Hình 2.8 Sơ đồ lớp CCActionIntervel 16](#_Toc439087872)

[Hình 2.9 Sơ đố các lớp kế thừa từ CCActionEase 17](#_Toc439087873)

[Hình 2.10 Sơ đồ các lớp kế thừa từ CCTilesGrid3Daction 18](#_Toc439087874)

[Hình 2.11 Sơ đồ các lớp kế thừa từ CCGrid3DAction 18](#_Toc439087875)

[Hình 2.12 Hiệu ứng lật trang 19](#_Toc439087876)

[Hình 2.13 Sơ đồ các lớp kế thừa từ CCActionInstant 19](#_Toc439087877)

[Hình 2.14 Logo giấy phếp GNU 41](#_Toc439087878)

[Hình 2.15 Logo giấy phép Apache 54](#_Toc439087879)

[Hình 3.1 Sơ đồ Use-case 63](#_Toc439087880)

[Hình 3.2 Sơ đồ màn hình 67](#_Toc439087881)

[Hình 3.4 Màn hình bảng xếp hạng 69](#_Toc439087882)

[Hình 3.5 Màn hình khi có sẵn map 71](#_Toc439087883)

[Hình 3.6 Màn hình thêm mới map 72](#_Toc439087884)

[Hình 3.7 Màn hình chỉnh sửa map 73](#_Toc439087885)

[Hình 3.8 Màn hình chơi game chính 75](#_Toc439087886)

[Hình 3.9 Màn hình chơi game khi tạm dừng 76](#_Toc439087887)

[Hình 3.10 Màn hình chơi game khi hoàn thành thử map 77](#_Toc439087888)

[Hình 4.1 Màn hình chơi game không sử dụng quadtree 85](#_Toc439087889)

[Hình 4.2 Màn hình chơi game sử dụng quadtree 85](#_Toc439087890)

[Hình 4.3 Màn hình chỉnh sửa map 86](#_Toc439087891)

[Hình 4.4 Màn hình bảng xếp hạng map 87](#_Toc439087892)

DANH MỤC BẢNG

Bảng 2.2.2 Các tính năng của Jetty 24

Bảng 2.2.3 So sánh các application server 25

Bảng 2.3.3 So sánh các nosql database 26

Bảng 2.1 So sánh một số giấy phép mã nguồn mở 60

Bảng 3.1 Danh sách Actor 63

Bảng 3.2 Danh sách use-case 64

Bảng 3.3 Đặc tả use-case create map 64

Bảng 3.4 Đặc tả use-case modify map 65

Bảng 3.5 Đặc tả use-case upload map 65

Bảng 3.6 Đặc tả use-case play game 66

Bảng 3.7 Đặc tả use-case rate map 66

Bảng 3.8 Bảng chú thích cho màn hình chính 68

Bảng 3.9 Bảng chú thích cho màn hình bảng xếp hạng 70

Bảng 3.10 Bảng chú thích cho màn hình khi có sẵn map 71

Bảng 3.11 Bảng chú thích cho màn hình thêm mới map 72

Bảng 3.12 Bảng chú thích cho màn hình chỉnh sửa map 74

Bảng 3.13 Bảng chú thích cho màn hình chơi game chính 75

Bảng 3.14 Bảng chú thích cho màn hình chơi game khi tạm dừng 76

Bảng 3.2.1 So sánh các kiến trúc server 78

Bảng 3.2.4 Bảng so sánh set, sorted set, list, vector 82

DANH MỤC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| TỪ VIẾT TẮT | TỪ ĐẦY ĐỦ | ĐỊNH NGHĨA TIẾNG VIỆT |
| 2D | Two - dimensional space | Không gian hai chiều |
| GDI | Graphics Device Interface | Giao diện đồ họa thiết bị |
| GPL | General Public License | Giấy phép cộng đồng chung |
| BSD | Berkeley Software Distribution | Giấy phép phân phối phần mềm ở Berkeley |

TÓM TẮT KHÓA LUẬN

Đề tài được nguyên cứu xoay quanh trọng tâm chính là hệ thống challenge game Speed Runner. Hệ thống bao gồm 4 thành phần là:

+ Một game thể loại runner đa nền tảng

+ Map editor tool cho phép người dùng tạo map mới, upload, chia sẻ với bạn bè

+ Webserver làm nhiệm vụ tiếp nhận các request upload cũng như download map.

+ Database server để lưu trữ các map đã được upload lên

Với SpeedRunner, người chơi có thêm lựa chọn game mới, không những thế , họ còn có thể tạo ra những map mới theo phong cách riêng của họ.

LỜI MỞ ĐẦU

Ngành công nghiệp game đã và đang phát triển không ngừng, nhờ sự tiến bộ và phát triển của khoa học công nghệ, sản phẩm công nghệ cao. Từng giây trôi qua, các studio game cạnh tranh khốc liệt nhằm tạo ra các sản phẩm chất lượng với những ý tưởng sáng tạo đột phá, dường như bất kì ý tưởng nào đều được các studio tận dụng triệt để.

Tuy nhiên, dường như các studio đã quá “tham lam” khi giành hết phần việc sáng tạo về phía mình mà không cho người chơi tham gia vào quá trình sáng tạo.

Luận văn “Xây dựng và phát triển hệ thống challenge game trên nền tảng thiết bị di động với Speed Runner”, với mong muốn cung cấp 1 hướng đi mới, giúp người chơi thỏa chí sáng tạo cũng như giảm bớt gánh nặng cho studio phát triển game.. Đề tài góp phần làm cho quá trình phát triển game linh hoạt hơn, phổ thông hơn, chiến lược hơn và cộng đồng hơn. Kết quả nguyên cứu được chính là một hệ thống hỗ trợ phát triển game EasyDesign, liên kết với framework eDesign xây dựng trên Cocos2d-x, hỗ trợ tính năng kéo thả và tạo ra một game tương đối hoàn thiện và có khả năng tiếp tục phát triển mã nguồn.

Nội dung được trình bày trong 6 chương. Cụ thể, Chương I trình bày khái quát về đề tài. Chương II đề cập đến các lý thuyết cơ sở liên quan đến đề tài như lý thuyết về mô hình thiết kế IDE, Cocos2d-x, phần mềm mã nguồn mở và trình bày các phương pháp nguyên cứu đã được áp dụng trong việc nguyên cứu đề tài. Chương III trình bày chi tiết quá trình phân tích và thiết kế hệ thống Easy Design. Chương IV Trình bày về framework eDesign. Chương V Kết quả thực nghiệm và đánh giá của thị trường được nghiên cứu đối với sản phẩm. Chương VI Kết luận - Hướng phát triển được trình bày nhằm đúc kết những thiếu sót của đề tài đối với các vấn đề hiện tại đang giải quyết.

# GIỚI THIỆU VỀ ĐỀ TÀI

## *LÝ DO CHỌN ĐỀ TÀI.*

Video game đã xuất hiện từ những năm 1950, nhưng đến những năm 1960 thì ngành công nghiệp game mới đặt nền móng cho việc phát triển sau này. Dù trong giai đoạn phát triển của ngành công nghiệp non trẻ, ngành công nghiệp Game vẫn khẳng định mình trong giai đoạn giải trí của con người. Bắt đầu phát triển và trở nên thịnh vượng từ năm 1950 đến đầu những năm 1970, suy sụp đầu tiên năm 1977 và cơn khủng hoảng kinh hoàng năm 1983. Sau đó, ngành công nghiệp này đã có nhiều bước chuyển biến tích cực trên thị trường và vương lên chiếm thị phần của các ngành công nghiệp già cỗi khác.

Tính đến đầu thế kỷ 21, nhiều thể loại game và dòng game đã được ra mắt thị trường với nhiều sản phẩm nổi bật. Tuy đa dạng về thể loại game và số lượng game, tuy nhiên, đại đa số các game đều có 1 điểm chung là bó buộc người chơi theo 1 kịch bản sẵn có, điển hình nhất là dòng game platform, điều này đôi khi gây cho người chơi cảm giác ức chế, chán nản. Tuy nhiên, cũng có 1 số game cho phép người chơi sáng tạo, làm ra cái mới, có thể kể đến như

+ Minecraft, cho phép người chơi có thể tạo ra mọi thứ trên đời từ phi thuyền không gian, tới cả 1 thành phố, thậm chí là tựa game pacman huyền thoại

+ Warcraft, với hệ thống map editor mạnh mẽ đi cùng, người chơi đã có thể tạo ra rất nhiều custom map khác nhau, điển hình nhất là DOTA, tựa game đánh dấu sự ra đời của dòng game rất được ưa thích hiện tại MOBA

+ Super Mario maker: hệ thống tạo và chia sẻ map của game mario trên hệ máy Wii mới ra tháng 9/2015

Các game trên đều đạt được những thành công vượt trội, cho thấy rằng việc cho phép người chơi sáng tạo trong game là 1 điều đáng để cân nhắc.

Ngoài ra, có 1 điều đáng để lưu tâm nữa đó là: nền tảng phát triển. Trong 3 game vừa kể trên, ngoài minecraft có thể chơi trên nhiều nền tảng, thì warcraft do đặc trưng thể loại game nên hầu như chỉ có thể chơi trên PC, trong khi Super Mario maker lại chỉ chơi được trên Wii; điều này vô hình chung lại là 1 điểm trừ lớn khi mà trong những năm gần đây, các thiết bị di động( điện thoại di động, máy tính bảng và các thiết bị di động khác) đã và đang phát triển mạnh mẽ với những con số ấn tượng: 4 tỉ người sử dụng điện thoại di động, 86% người sử dụng điện thoại di động truy cập Internet trong khi đang xem TV, theo dự đoán, năm 2015, lượng người sử dụng điện thoại di động để truy cập Internet sẽ vượt qua lượng người dùng máy tính cá nhân.

Với ưu thế nhỏ gọn có thể giúp người dùng giải trí khắp mọi nợi, theo thống kê, số người sử dụng điện thoại di động để chơi game chiếm tới 61%, dẫn tới nhu cầu chơi game trên thiết bị di động cực kì lớn.

Từ những điều trên, nhận thấy việc xây dựng game đa nền tảng hỗ trợ người chơi sáng tạo, đang là 1 khía cạnh còn nhiều trống vắng, với mong muốn đóng góp cho nền công nghiệp game cũng như ứng dụng các kiến thức đã học, nhóm quyết định thực hiện đề tài này.

## *MỤC TIÊU ĐỀ TÀI*

Mục tiêu của đề tài là xây dựng 1 game đa nền tảng gắn kèm theo hệ thống map editor cho phép người chơi tự tạo map, lưu trữ cũng như chia sẻ, thách thức bạn bè của họ.

## Ý *NGHĨA ĐỀ TÀI*

### Đối với người chơi nói riêng và ngành công nghiệp game nói chung

Tạo ra 1 game mới, làm tăng phần phong phú trong kho game hiện tại.

### Đối với bản thân nhóm

* Giúp trau dồi đồng thời nâng cao kiến thức về lập trình
* Ứng dụng những gì đã học được vào quá trình làm game
* Có được những kinh nghiệm quý báu về làm game client - server

## Đối tượng nghiên cứu:

* Một runner-game và map editor xây dựng trên ngôn ngữ C++ và nền tảng cocos2dX.
* Webserver xây dựng trên ngôn ngữ Java sử dụng Jetty.
* Database server được xây dựng bằng Redis .

## Phạm vi nghiên cứu:

### Ngôn ngữ C++ và engine Cocos2d-x:

* Tìm hiểu về cấu trúc và kỹ thuật của engine Cocos2dx-x thực thi các hành động, hiệu ứng, xử lý thông dụng và cần thiết cho game trên di động, máy vi tính.
* Tìm hiểu và ứng dụng các kỹ thuật lập trình hướng đối tượng và cấu trúc ngôn ngữ C++, cùng các bộ thư viện cơ bản.
* Nghiên cứu các cấu trúc xây dựng một game client- server thông dụng và ứng dụng của các cấu trúc framework game của các công trình đi trước
* Nghiên cứu quadtree vào việc tối ưu performance

### Ngôn ngữ Java và Jetty serverlet container:

* Tìm hiểu các Java util class như Map, Set, List, Random
* Nghiên cứu cách tạo 1 web server với Jetty
* JSON

### Redis

* Nghiên cứu cấu trúc Redis nosql database server
* Nghiên cứu và ứng dụng Jedis, redis client cho java

# CƠ SỞ LÝ THUYẾT

## Cocos2d-x.

### Giới thiệu

Theo ông Gustavo E Rangel, trong cuốn “Cocos2d-X 3.0: Game Development Programming for iOS and Android” xuất bản năm 2014, Cocos2d-x là một engine mã nguồn mở đa nền tảng dưới giấy phép MIT được phát triển dựa trên nền tảng engine Cocos2d. Cocos2d-x được sử dụng để xây dựng các trò chơi, và ứng dụng đa nền tảng như iOS, Andriod, Windows Phone và nhiều nền tảng khác. Người lập trình có thể sử dụng ngôn ngữ C++, JavaScript và Lua để phát triển ứng dụng. Engine đã làm việc rất hiệu quả; ít nhất 25% game di động trên thế giới được hỗ trợ bởi Cocos2d-x. Ưu điểm chính của engine là chỉ với một mã nguồn duy nhất, được chuyển sang nhiều nền tảng khác nhau mà không cần tốn bất kỳ chi phí nào khác.

Cocos2d-x được thiết kế và xây dựng trên lớp cấu trúc kế thừa chặt chẽ chạy trên nền OpenGL 2.0 và OpenGL ES 2.0. Với các thành phần tiêu biểu, Lớp CCScene tạo nên một cảnh trong game, hệ thống các cảnh tạo nên cấu trúc game. CCDirector Lớp singleton, được truy xuất toàn cục bằng cái gọi getInstance, dùng để quản lý các cảnh (CCScene) trong game. CCNode yếu tố cơ bản của Cocos2d-x, là một lớp trừu tượng và là lớp cơ sở cho tất cả các nút trong cấu trúc Cocos2d-x như: CCScene, CCLayer, CClabelTTTF, CCSprite, CCMenuItem. CCAction Lớp được sử dụng để thực hiện các thao tác di chuyển, thay đổi kích thước, xoay, làm mờ và nhiều hiệu ứng khác trên một nút. CCTransitionScene giúp chuyển đổi giữ các cảnh trở nên đẹp hơn và chuyên nghiệp hơn.

### Giấy phép

Dựa trên tài liệu được công khai tại trang chính thức của Cocos2d-x - [http://www.cocos2d-x.org/reference/native-cpp/V3.3/index.html](http://www.cocos2d-x.org/reference/native-cpp/V3.3/index.html%20năm%202014) năm 2014: “Bản quyền © 2010-2012 cocos2d-x.org

Giấy phép này được cấp, hoàn toàn miễn phí, để bất kỳ người nào cũng có thể có được một bản sao của phần mềm này và các tài liệu liên quan (“Phần mềm”), để không bị hạn chế khi làm việc với phần mềm, bao gồm các quyền không giới hạn về việc sử dụng, sao chép, sửa đổi, sáp nhập, xuất bản, phân phối, cấp phép, và / hoặc bán các bản sao của phần mềm, và cho phép người đã trang bị phần mềm thực hiện các quyền trên, nhưng phải tuân theo các điều kiện sau đây:

Thông báo bản quyền ở trên và Giấy phép thông báo này phải được kèm trong tất cả các bản sao hoặc phần quan trọng của phần mềm.

PHẦN MỀM ĐƯỢC CUNG CẤP “NHƯ LÀ” (“AS IS”), KHÔNG ĐẢM BẢO VỚI MỌI HÌNH THỨC, RÕ RÀNG HAY NGỤ Ý, BAO GỒM KHÔNG GIỚI HẠN CÁC ĐẢM BẢO THƯƠNG MẠI, PHÙ HỢP VỚI MỘT MỤC ĐÍCH CỤ THỂ VÀ KHÔNG VI PHẠM. TRONG BẤT CỨ TRƯỜNG HỢP NÀO TÁC GIẢ HOẶC NGƯỜI GIỮ BẢN QUYỀN SẼ KHÔNG CHỊU TRÁCH NHIỆM CHO BẤT CỨ YÊU CẦU, THIỆT HẠI HOẶC TRÁCH NHIỆM KHÁC, DÙ TRONG MỘT HOẠT ĐỘNG HỢP ĐỒNG, SAI PHẠM (TORT) HOẶC NHỮNG VẤN ĐỀ KHÁC, PHÁT SINH TỪ BÊN NGOÀI HOẶC LIÊN QUAN ĐẾN PHẦN MỀM HAY SỬ DỤNG HAY NHỮNG HOẠT ĐỘNG KHÁC TRONG PHẦN MỀM.”

### Các tính năng

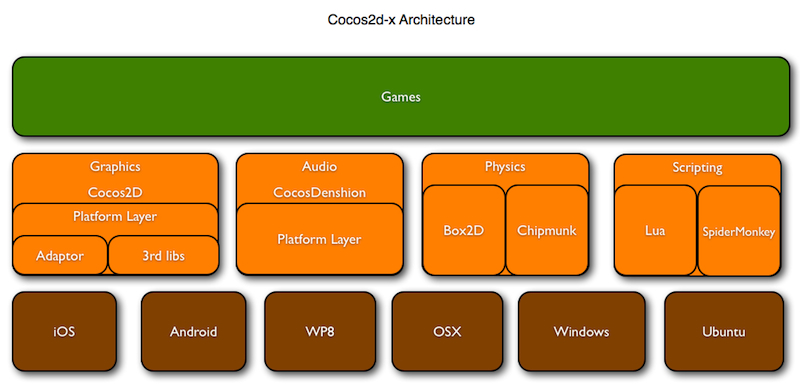
Cocos2d-x được sử dụng bởi các cá nhân và các tổ chức được biết như các công ty lớn. Ngày nay, nhiều game được viết bằng Cocos2d-x đang thống trị bảng doanh thu cao nhất của AppStore và Google Play, đặc biệt tại các quốc gia: Trung Quốc, Hàn Quốc và Nhật Bản. Những kỹ sư từ ChuKong, Google, Microsoft, ARM, Intel và BlackBerry đang hoạt động rất tích cực trên cộng đồng Cocos2d-x. Những ông chủ lớn như Zynga, Wooga, Glu, IGG, Big Fish Games, Fingersoft, Gamevil, GREE, DeNA, Konami, CJ E&M, NHN, LINE, Square Enix, Disney mobile đã sử dụng Cocos2d-x để phát triển game của mình.

Nguyên nhân chính để chọn Cocos2d-X cho luận văn bởi các yếu tố:

* API hiện đại dành cho C++.
* Đa nền tảng – cả di động và máy để bàn.
* Có thể kiểm tra và tìm lỗi của game trên máy tính để bàn trước khi thật sự đưa nó đến với các đối tượng di động hoặc máy tính để bàn hướng đến.
* Hệ thống API lớn với đầy đủ chức năng bao gồm hình cho game (sprites), hành động, diễn hình (animation), biễu diễn hạt (particles), chuyển (transitions), thời gian (timers), sự kiện (events) ( chạm, bàn phím, cảm biến gia tốc, chuột), âm thanh, Nhập xuất tập tin, chuyển động có khung xương, 3D.
* Khung xương chuyển động
* Kiểu chữ:
* Vẽ kiểu chữ nhanh sử dụng cơ chế cố định (fixed) và chiều dài cho kiểu chữ
* Hỗ trợ nhiều kiểu font chữ và định dạng .ttf
* Hỗ trợ Bản đồ theo ô (Tile Map): Tứ phương, tạo hiệu ứng 3D (Isometric) và lục phương.
* Cuộn màn hình
* Chuyển động đường sọc
* Tạo Texture
* Chạm và cảm biến gia tốc trên di động
* Chạm, Chuột, Bàn phím trên máy tính để bàn
* Engine hỗ trợ âm thanh (thư viện CocosDeshion) dựa trên OpenAL
* Tích hợp chuyển động chậm/ Nhanh về phía trước
* Nhanh và nén Texture
* PVR nén và giải nén texture
* ETC1 nén texture và nhiều hơn nữa
* Giải pháp độc lập
* Giao diện tiện ích cho người dùng (UI Widgets).
* Bố trí theo trục tung và trục hoành
* Các đối tượng chứa: ListView, PageView, ScrollView
* Tiện ích: Nút, Khung đánh dấu, Thanh tải, Thanh trượt, khung chữ
* Vật lý.
* Tích hợp Chipmunk2D bên trong
* Tích hợp Box2D bên trong
* Dễ sử dụng với lớp bọc hướng đối tượng với cơ thể, tiếp xúc, điểm nối, các khối hình
* Âm thanh.
* Hiệu ứng âm thanh
* Âm thanh nền
* Mạng.
* HTTP với SSL
* WebSocket API
* XMLHttpRequest API

### Kiến trúc engine Cocos2d-x

Kiến trúc Cocos2d-x - www.cocos2d-x.org/wiki/Engine\_Architecture (2014):



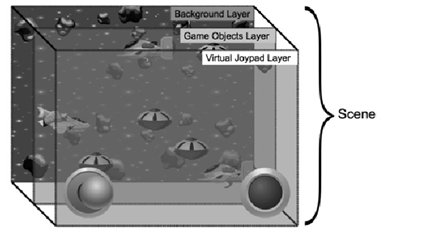
Hình 2.1 Kiến trúc Cocos2d-x

Kiến trúc trên thể hiện tính đa nền tảng của Cocos2d-x với khả năng chạy trên iOS, Android, WP8, OSX, Windows, Ubuntu nhờ các tầng adaptor và 3rd libs được bao bọc bởi các Platform Layer. Riêng về các yếu tố như vật lý và mã kịch bản, Cocos2d-x được hỗ trợ bởi Box2D và chipmunk hay lua và spiderMonkey nên thuận tiện cho việc đa nền tảng.

##### CCScene

Tất cả các đối tượng hiển thị trong game, cocos2d-x đều xem đó như là một nút và các nút đó được kế thừa bởi lớp CCNode. Để hiển thị bất cứ đối tượng nào trong game như hình (sprite) hay chữ (label) thì chúng phải được xem như là một nút và được quản lý bởi nút gốc là CCScene.

Giả sử như đây là một màn hình khi chơi game:



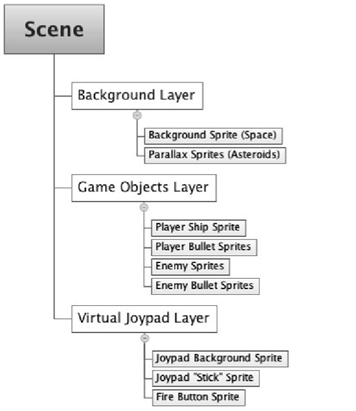
Hình 2.2 Khám phá các lớp của Cocos2d-x



Hình 2.3 Chụp màn hình của một game Cocos2d-x

Như trong hình, cảnh game này hiện tại có 3 lớp chính là Nền (Background), Đối tượng (Game Objects) và Lớp điều khiển (Virtual Joypad). Trong đó để hiển thị các thiên thạch ở lớp nền, các phi thuyền và điều khiển thì CCScene trước tiên được tạo ra và quản lý bởi CCDirector. Sau đó, CCScene sẽ thêm các nút con là Lớp nền, Lớp đối tượng, Lớp điều khiển. Các lớp này lần lượt thêm các nút con là hình thiên thạch cho lớp nền, hình phi thuyền cho lớp đối tượng và hình khung điều khiển cho lớp điều khiển.

Sơ đồ cây của lớp Scene:



Hình 2.4 Sơ đồ cây của một màn

Sử dụng nhiều lớp trong một cảnh cũng giúp quản lý các nút trở nên dễ dàng hơn. Người dùng có thể ẩn hoặc di chuyển đi toàn bộ nút trong cùng cảnh. Sắp xếp lại các lớp được tạo nên thứ tự ưu tiên khi vẽ và thậm chí khi xoay hoặc thay đổi tỉ lệ cũng sẽ ảnh hưởng đến toàn bộ các nút bị nút đó ảnh hưởng.

Cocos2d-x không quy định một cảnh (scene) phải có lớp mà có thể dùng một nút (Node) để thay cho lớp (layer) nhưng tốt nhất nên có ít nhất có 1 lớp (layer) được chứa bởi 1 cảnh (scene) và lớp này sẽ chứa các nút khác như hình, chữ, khung chọn, hộp thoại, vv.

Mỗi nút đều có thể chứa 1 nút khác và mỗi nút trong hệ thống cây, trừ nút cảnh (scene), đều có một nút cha. Các nút hoàn toàn quan hệ với nhau như một cây và không có quan hệ kế thừa đối với các nút cha, con.

Điểm khác biệt lớp nhất của CCScene với CCNode chỉ đơn giản chỉ là nút trừu tượng chỉ được dùng làm nút gốc. Và cũng như vậy, CCLayer chỉ đơn giản là có thêm các tương tác với các thành phần đầu vào như chạm, cảm biến giá tốc, ngoài ra, CCLayer không có điểm gì khác với CCNode.

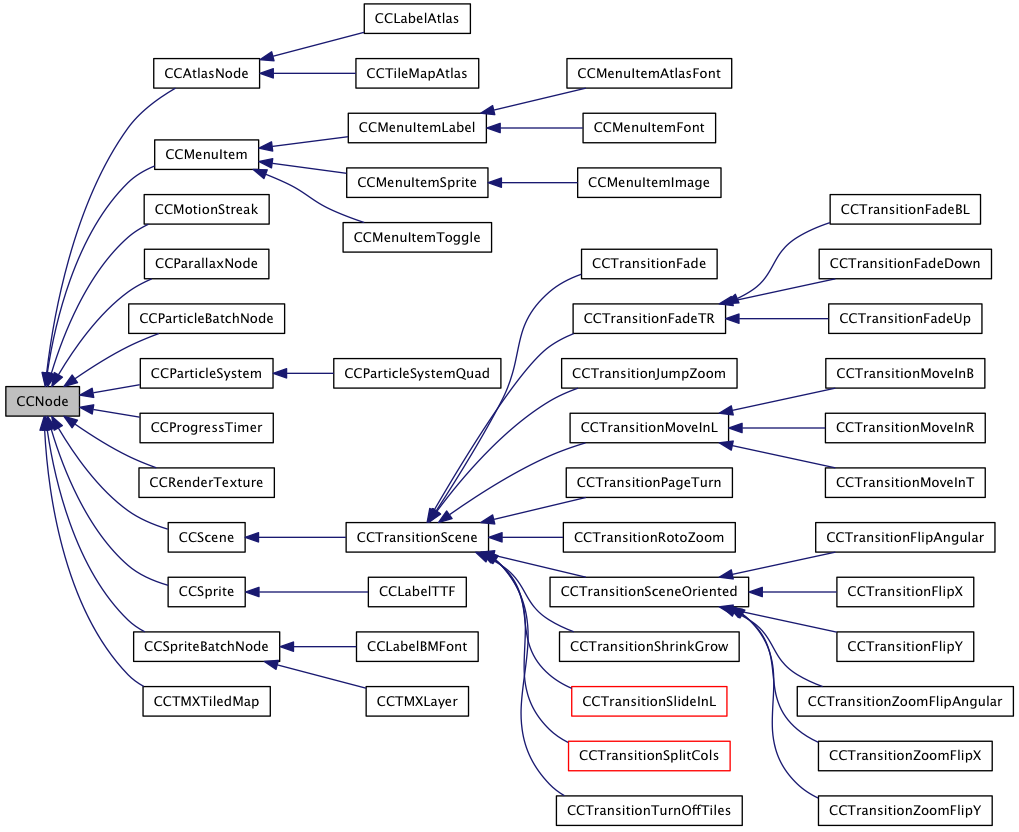
Tọa độ nút trong Cocos2d-X có liên quan đến tọa độ của nút cha, các nút con sẽ kế thừa các thuộc tính từ nút cha như tỉ lệ, độ xoay riêng màu sắc và độ hiển thị hình ảnh (Opacity, nằm trong khoảng 0 -> 255, tương ứng với alpha 0 -> 1).



Hình 2.5 Vị trí của một nút con khi được thêm vào nút cha

2.1.2.4.2 CCNode

Sơ đồ các lớp được kế thừa từ CCNode. Và trong đó, các lớp thường được sử dụng nhất được tô màu xám:



Hình 2.6 Sơ đồ các lớp kế thừa từ CCNode

CCNode là lớp cơ sở của tất cả các nút. CCNode là một lớp trừu tượng không chứa bất kì một hàm cụ thể nào và định nghĩa tất cả các thuộc tính và phương thức chung nhất cho tất cả các nút.

CCNode chứa tất cả các phương thức để quản lý (thêm, xóa, lấy đối tượng) cho các nút con.

**Khởi tạo một nút**:

CCNode \*childNode = CCNode::create();

**Thêm một nút**:

myNode->addChild(childNode, tag); // tag (kiểu int) dùng định danh cho nút đó gọi là “tag”.

**Lấy nút**:

myNode->getChildByTag(tag); // lấy childNode đã được định danh là tag (kiểu int)

**Xóa 1 nút bằng tag, dọn sạch toàn bộ các hành động đang chạy**:

myNode->removeChildByTag(tag, true); // true là dọn sạch

**Xóa tất cả các** **nút**:

myNode->removeAllChild();

2.1.2.4.3 CCAction

Một node có thể chạy các hành động. Các hành động (Action) là một lớp được sử dụng để thực hiện các thao tác di chuyển, thay đổi kích thước, xoay, làm mờ đi và nhiều hiệu ứng khác nhau trên một nút. Bởi vì lớp hành động này làm việc trên mọi nút, nên chúng ta có thể sử dụng chúng lên các hình (sprite), hay chữ (label) và thậm chí là các khung chọn (menu) hoặc toàn bộ một cảnh (scene).

**Khởi tạo một hành động**:

CCAction \*action = CCMoveBy::create(0.1f, 10); // CCMoveBy là một hành động, dùng để di chuyển

**Chạy các hành động**:

myNode->runAction(action, tag); // tag (kiểu int) để định danh cho hành động

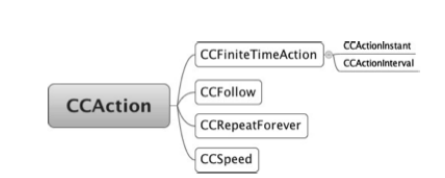
**Ngừng một hành động bằng định danh tag**:

myNode->stopActionByTag(tag);

**Ngừng toàn bộ các hành động của một nút**:

myNode->stopAllAction();

Dưới đây là sơ đồ các lớp kế thừa từ lớp CCAction.



Hình 2.7 Sơ đồ lớp kế thừa từ CCAction

Có 3 lớp Hành động kế thừa trực tiếp từ CCAction:

* CCFollow: cho phép một nút theo sau một nút khác
* CCRepeatForever: lặp lại một nút vô tận
* CCSpeed: thay đổi tần suất cập nhật của một hành động khi nó đang chạy

**CCFollow** có thể thiết lập một nút có hành động giống một nút khác. Ví dụ, đối tượng tên nhân vật (chữ - label) chạy đính vào một người chơi (player)

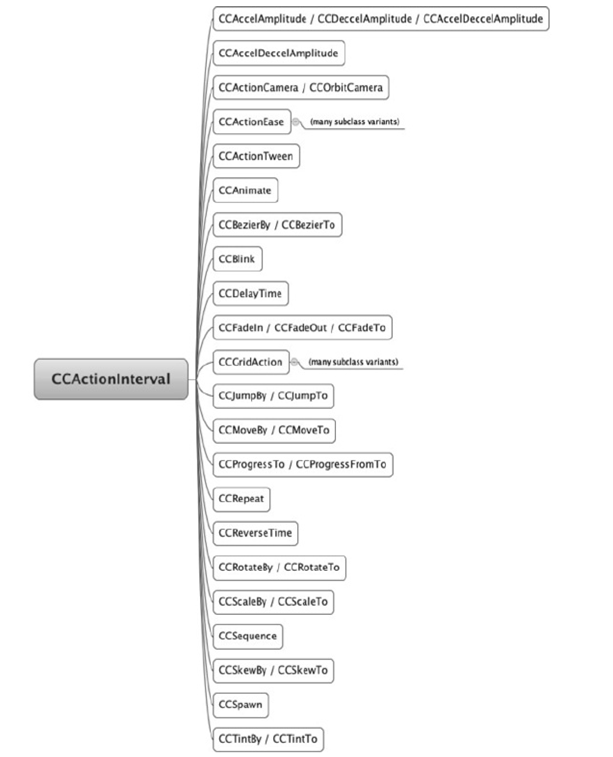
Label->runAction(CCFollow::create(player));

CCRepeatForever cho phép tạo ra chuỗi các hành động nối nhau:

CCAction \*action = CCRepeatForever::create( CCMoveBy::create(0.1f, point1), CCMoveTo::create(0.1f, point2)); //point1, point2 là một điểu kiểu CCPoint

**CCSpeed** được dùng để thay đổi tốc độ của một hành động khác chạy, dùng tương tự như các hành động khác với lệnh create và runAction. Nhưng, CCSpeed không thể thêm vào trong CCSequence.

**Interval** được tạo ra để giúp thực hiện các hành động diễn ra trong một khoảng thời gian nhất định và hành động đó không cần xóa khi thực hiện xong các hành động sẽ tự xóa.



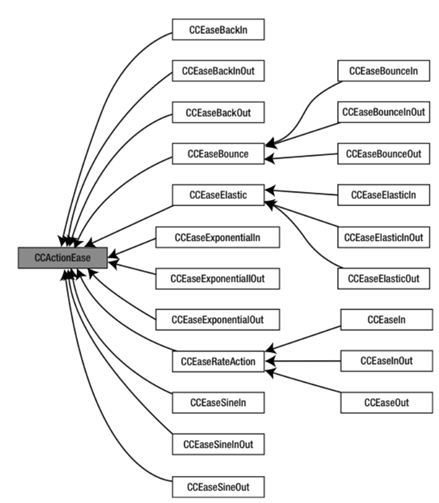
Hình 2.8 Sơ đồ lớp CCActionIntervel

**CCSequence** cho phép một hành động thực hiện sau khi một hành động khác được thực hiện xong. CCSequence cũng có thể sử dụng CCRepeatForever, CCSpeed. Lưu ý, Cocos2d-x cần thêm NULL vào sau các hành động trong CCSequence, do đó, nếu không để NULL ở cuối chuỗi hành động sẽ bị lỗi.

**Ease** cho phép thay đổi hiệu ứng của Hành động mọi lúc. Ease tạo ra các chuyển động mà thông thường tốn rất nhiều thời gian để viết một đoạn mã thực thi. Thuộc tính rate xác định mức độ hiệu ứng và phải lớn hơn 1 mới thấy được hiệu ứng.

CCMoveTo \*move = CCMoveTo::create(11, ccp(11, 1992));

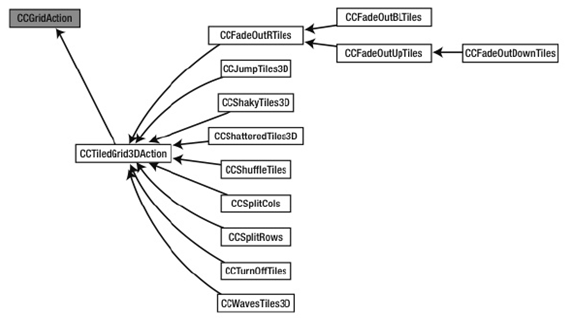
myCode->runAction(CCEaseInOut::create(move,4));



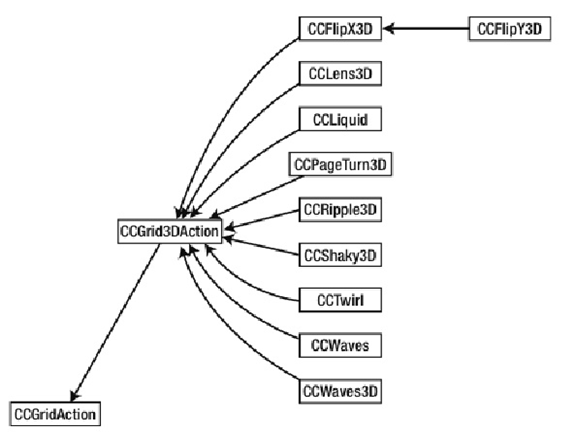
Hình 2.9 Sơ đố các lớp kế thừa từ CCActionEase

**Grid** có 2 lớp con là CCDrid3Daction và CCTiledGrid3Daction. Dưới đây là sơ đồ của CCGridAction:

Grid tạo nên các hiệu ứng 3D nhưng hiệu ứng sẽ không thực hiện được nếu không được bật depth buffering, điều này sẽ làm tốn bộ nhớ rất nhiều và chi phí xử lý.



Hình 2.10 Sơ đồ các lớp kế thừa từ CCTilesGrid3Daction



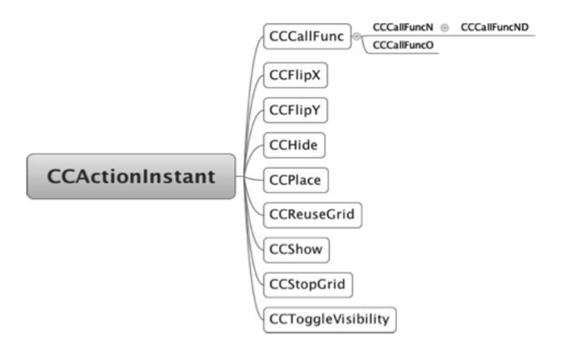
Hình 2.11 Sơ đồ các lớp kế thừa từ CCGrid3DAction

Để thiết lập depth buffering, một số đoạn mã trong lớp ủy nhiệm (Delegate) lúc khởi tạo EAGLView, mặc định thuộc tính DepthFormat là 0, để thiết lập depth buffering chúng truyền vào các đối số GL\_DEPTH\_COMPONENT\_OES cho 16-bit depth buffer hoặc GL\_DEPTH\_COMPONENT24\_OES cho 24-bit depth buffer.



Hình 2.12 Hiệu ứng lật trang

**Instant** được tạo ra để hỗ trợ cho các chuỗi hành động. Đôi khi trong chuỗi các hành động phải thay đổi một thuộc tính của nút, sau đó tiếp tục các hành động khác.



Hình 2.13 Sơ đồ các lớp kế thừa từ CCActionInstant

Điểm khác biệt giữa các lớp trên là loại chọn lọc (selector) nào sẽ được gọi và ngữ cảnh nào mà phương thức được gọi.

2.1.2.4.4 Scheduled

Lịch trình (scheduled) được cocos2d-x dùng để gọi thực thi các hàm. Dùng hủy lịch trình (Unschedule) để ngừng thực thi các hàm tương tự như lên lịch trình.

**ScheduleUpdate** dùng để gọi hàm Update. Update dùng để cập nhật trạng thái của nút mỗi khung (frame) tối đa nên là 60 lần 1 giây. Ví dụ:

myNode->scheduleUpdate();

Trong mã phải khai báo Cập nhật (Update) như sao:

void update(float dt) { ... }

**Schedule** dùng để gọi một hàm chọn lọc (selector) bất kỳ được khai báo giống với hàm lịch trình (Schedule Update) chỉ có điểm khác là có thời gian chờ trong các lần gọi lịch trình. Nếu truyền số vào là 0 thì nó giống như hàm lịch trình (ScheduleUpdate).

myNode->Schedule(selector, 3); // 3 là thời gian giữa 2 lần gọi schedule

void selector(float dt) { ... }

**ScheduleOnce** dùng để gọi hàm chỉ một lần không lặp dùng tương tự như hàm lịch trình (schedule) chỉ có điểm khác là số được truyền vào là thời gian tính từ lúc gọi đến lúc thực thi chọn lọc (selector).

**ScheduleUpdateWithPriority** Nếu nhiều nút cùng gọi hàm lịch trình (schedule update) thì hàm nào có độ ưu tiên càng nhỏ sẽ được thực thi trước (mặc định là 0).

2.1.2.4.5 CCDirector

CCDirector là một phần quan trọng của cocos2d-x. Lớp CCDirector là một lớp đơn (singleton), nghĩa là chỉ có duy nhất một thể hiện tại mọi thời điểm và CCDirector có thể được truy xuất toàn cục bằng cách gọi sharedDirector hay getInstance.

CCDirector chứa các thiết lập toàn cục cho cocos2d-x và quản lý các cảnh (scene) trong cocos2d-x. Một số chức năng của lớp CCDirector:

* Truy xuất vào cảnh hiện tại của màn hình.
* Chạy, thay thế, thêm, bớt một cảnh.
* Truy xuất vào các thiết lập của cocos2d-x.
* Truy xuất vào OpenGL view và hệ điều hành window trong cocos2d-x.
* Hỗ trợ ngừng, chạy lại và kết thúc trò chơi.
* Xác định cách mà trạng thái game được Cập nhật (Update).

Để chạy trò chơi, đầu tiên phải khởi tạo một cảnh (scene) nằm ở cuối hàm applicationDidFinishLaunching trong AppDelegate. Dùng CCDirector để bắt đầu chạy với hàm runWithScene

CCDirector::getInstance()->runWithScene(scene); // phiên bản 3.3

Thay thế bằng 1 cảnh khác (otherScene):

CCDirector::getInstance()->replaceScene(otherScene);

Dùng hiệu ứng để chuyển cảnh (changeScene):

CCSceneTransition tran = {loại chuyển màn hình}::create(2, changeScene); // 2 là số giây trước lúc chuyển

CCDirector::getInstance()->replaceScene(otherScene);

Lưu ý, khi thay thế một cảnh mới cảnh cũ sẽ bị xóa hoàn toàn khỏi RAM và các nút, lịch trình cũng bị hủy.

**Pushing, Poping** cho phép thay thế cảnh mới vào mà không phải giải phóng cảnh cũ, Giúp chuyển qua các cảnh mới nhanh hơn. Tuy nhiên, các cảnh sẽ chiếm một lượng RAM khác lớn có thể dẫn đến lỗi khi chơi.

Một số vấn đề khác, khi đẩy một cảnh (push) vào phải nhớ số lượng cảnh để có thể lấy ra (pop) một cách chính xác. Nếu không khi lấy cảnh sẽ xảy ra lỗi và gãy trò chơi.

### Những điểm mạnh của cocos2dx:

* Cross-platform: ngày nay hầu hết các framework đều cross-platform. Nhưng vấn đề là Cocos2d-x làm được nhiều hơn hẳn, cụ thể, Cocos2d-x hỗ trợ cho: windows, mac, linux, ios, android, blackberry, tizen, winRT, window phone
* Open source: hoàn toàn open source, có thể làm bất kỳ điều gì với code của nó đồng thời cũng có thể đóng góp vào framework nữa để giúp nó phát triển hơn.
* Light weight & Fast running: sử dụng C++, Cocos2d-x có hiệu suất cao, chạy được trên cả những thiết bị cấu hình rất thấp của Android.
* APIs đơn giản và dễ học
* Lua binding, Js binding: hoàn toàn có thể sử dụng Lua, hay javascript để hiện thực sản phẩm, (không nhất thiết là C++). Điều này rất hữu ích đối với những người mới, không mạnh về C++
* Tools và Library: hiện tại có rất nhiều công cụ và thư viện hỗ trợ phát triển game bằng Cocos2dx, có thể xem tại trang hub:

<http://www.cocos2d-x.org/hub>

Nổi bật trong số đó là cocos studio

### So sánh với các 2d game engine khác

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Cocos2dx | GameMaker | MonoGame | AndEngine | GameSalad | Love |
| Giá thành | Hoàn toàn miễn phí | Có bản miễn phí + trả phí | Chỉ miễn phí trên window | Hoàn toàn miễn phí | Trả phí/ miễn phí | Hoàn toàn miễn phí |
| Open source | Có | Không | Có | Có | Không | Có |
| Ngôn ngữ sử dụng | C++, lua, javascript | Không viết code | C# | Java | Không viết code | Lua |
| Performance | Tốt | Tương đối | Tệ | Tương đối | Tương đối | Tương đối |
| Nền tảng hỗ trợ | Mac OS X,  IOS  Android,  Window phone,  Windows,  Linux,  Blackberry  Tizen,  WinRT | HTML5, Linux, Windows, Mac, Windows Phone, Android, iOS, PS3/4/Vita | Windows, Mac, Linux, iOS,  Android | Android | Windows 8, Android, iOS, HTML5, Mac OS X | Windows, Mac OS X, Linux |
| Mức độ dễ sử dụng | Dễ sử dụng | Không đòi hỏi kĩ năng lập trình, rất dễ sử dụng | Dễ | Dễ | Không đòi hỏi kĩ năng lập trình, rất dễ sử dụng | Tương đối |
| Cộng đồng | Lớn, thân thiện | Lớn | Ít | Tương đối | Lớn | Tương đối |
| Documentation | Đa phần đã lỗi thời | Rất đầy đủ | Ít | Không có tài liệu cụ thể | Tốt | Rất đầy đủ |
| Tutorial | Khá nhiều | Tốt | Tương đối | Ít | Tốt | Tốt |

## Jetty

### Giới thiệu

Jetty là 1 open source serverlet container, nó phục vụ các nội dung web trên nền tảng java như serverlet, JSP. Jetty được viết bằng Java và API của nó được cung cấp dưới dạng 1 tập hợp các file Jar. Người dùng có thể khởi tạo jetty như là 1 object và nhanh chóng thêm khả năng kết nối mạng vào 1 ứng dụng java.

Jetty bắt đầu nổi từ năm 1998, khi được release dưới giấy phép Apache 2.0

Website: http://www.eclipse.org/jetty/

### Các tính năng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Java Version | Protocols | Servlet Version | JSP Version |
| 9.3.x | 1.8 | HTTP/1.1, [HTTP/2](https://en.wikipedia.org/wiki/HTTP/2), [WebSocket](https://en.wikipedia.org/wiki/WebSocket) JSR356, [SPDY](https://en.wikipedia.org/wiki/SPDY) | 3.1 | 2.3 |
| 9.2.x | 1.7 | HTTP/1.1, [WebSocket](https://en.wikipedia.org/wiki/WebSocket) JSR356, [SPDY](https://en.wikipedia.org/wiki/SPDY) | 3.1 | 2.3 |
| 9.1.x | 1.7 | HTTP/1.1, [WebSocket](https://en.wikipedia.org/wiki/WebSocket) JSR356, [SPDY](https://en.wikipedia.org/wiki/SPDY) | 3.1 | 2.3 |
| 9.0.x | 1.7 | HTTP/1.1, [WebSocket](https://en.wikipedia.org/wiki/WebSocket), [SPDY](https://en.wikipedia.org/wiki/SPDY) | 3.0 (tracking 3.1 drafts) | 2.2 |
| 8.x | 1.6 | HTTP/1.1, [WebSocket](https://en.wikipedia.org/wiki/WebSocket), [SPDY](https://en.wikipedia.org/wiki/SPDY) | 3.0 | 2.1 |
| 7.x | 1.5, J2ME | HTTP/1.1, [WebSocket](https://en.wikipedia.org/wiki/WebSocket), [SPDY](https://en.wikipedia.org/wiki/SPDY) | 2.5 | 2.1 |
| 6.x | 1.4–1.5 | HTTP/1.1 | 2.5 | 2.0 |
| 5.x | 1.2–1.5 | HTTP/1.1 | 2.4 | 2.0 |
| 4.x | 1.2, J2ME | HTTP/1.1 | 2.3 | 1.2 |
| 3.x | 1.2 | HTTP/1.1 RFC2068 | 2.2 | 1.1 |
| 2.x | 1.1 | HTTP/1.0 RFC1945 | 2.1 | 1.0 |
| 1.x | 1.0 | HTTP/1.0 RFC1945 |  |  |

Bảng 2.2.2 Các tính năng của Jetty

### So sánh với các server application khác

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Jetty | JBoss | Tomcat | Liberty profile | Glassfish |
| Dễ cài đặt | 10 | 10 | 8 | 8 | 7 |
| Dễ cấu hình | 8 | 8 | 8 | 10 | 6 |
| Tài liệu | 7 | 10 | 8 | 8 | 7 |
| Giá thành | Miễn phí | Giới hạn | Miễn phí | Không | Giới hạn |
| Hiệu suất | 9 | 10 | 9 | 8 | 7 |
| Mở rộng | 10 | 8 | 9 | 8 | 9 |

Bảng 2.2.3 So sánh các application server

## Redis

### Giới thiệu

Redis là 1 open source, in-memory data structure store, được viết bằng ngôn ngữ ansi C, có thể sử dụng như 1 database, cache hoặc message broker. Redis hỗ trợ nhiều cấu trúc dữ liệu như: strings, hashes, lists, sets, sorted sets.

Ra mắt lần đầu: 10/4/2009

Giấy phép sử dụng: BSD.

Loại: nosql key-value store

Phiên bản ổn định: 3.0.6 (ra mắt ngày 18/12/2015)

### Đặc tính

* Điểm mạnh nhất: cực kì nhanh.
* Protocol: telnet.
* In-memory: có hỗ trợ back-up trên disk.
* Hỗ trợ cấu trúc dữ liệu: string, list, set, hash, sorted set.
* Cấu trúc dữ liệu đơn giản nhưng có thể thực hiện các lệnh phức tạp như .ZREVRANGEBYSCORE (trả về 1 danh sách các member trong 1 sorted set, sắp xếp bằng score từ cao xuống thấp..
* Có thể script bằng lua.
* Có transaction.
* Giá trị của key có thể đặt expire (tự động xóa sau 1 khoảng thời gian).
* Đa số các lệnh đều có time complexity là O(1).

### So sánh với các no-sql database khác

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Đặc điểm | Lưu trữ | Cấu trúc | Sử dụng khi | Mô tả |
| Redis | Cực nhanh | RAM | Key-value | Khi cần xử lý dữ liệu với tốc độ nhanh, nhưng khối lượng không thực sự lớn (do lưu trên ram) | Dữ liệu lưu trữ trên ram (có persistent), hỗ trợ các cấu trúc dữ liệu đơn giản và quen thuộc. hỗ trợ clustering + automatic failover |
| MongoDB | Khá giống sql | Disk | BSON | Truy vấn lượng lớn dữ liệu trên các field cố định | Có các tính năng quen thuộc từ sql (query, index) |
| Casandra | Dữ liệu cực lớn sql | Disk | CQL3 | Khi cần và phân tích lượng dữ liệu cực lớn | Lưu trữ và truy xuất lượng lớn với các thao tác quen thuộc của sql, tính scalability cao |
| CouchDB | Rất nhanh | Memcache | Key-value | Khi cần truy xuất lượng dữ liệu lớn, với độ trễ thấp và tốc độ cao | Dạng key-value, memcache, có hỗ trợ persistent và clustering |
| Neo4j | Graph database | Disk | Graph | Phù hợp cho việc xây dựng mạng xã hội, quản lý giao thông, bản đồ | Dữ liệu dạng đồ thị, được định nghĩa dựa trên các khái niệm quen thuộc trong lý thuyết đồ thị |

Bảng 2.3.3 So sánh các nosql database

## Quadtree

### Định nghĩa

Quadtree là một cấu trúc dữ liệu phân nhánh dạng cây, được sử dụng để phân hoạch vùng không gian hai chiều hiện tại thành các vùng nhỏ hơn và dễ quản lý hơn. Khác với cây nhị phân, mỗi lần phân hoạch, quadtree sẽ chia vùng hiện tại thành 4 vùng. Do đó mỗi node trong quadtree có thể có 4 node dẫn xuất hoặc không có node dẫn xuất nào, không có ngoại lệ.

Việc sử dụng Quadtree giúp chúng ta quản lý các đối tượng trong game một cách hiệu quả hơn, bằng cách phân chia vùng cần xử lý các đối tượng thành những vùng con, các đối tượng sẽ được đưa vào các vùng tương ứng và được quản lý riêng biệt. Quá trình phân chia cứ thế tiếp diễn cho đến khi mỗi vùng chỉ chứa một số lượng đối tượng nhất định hoặc mức độ phân chia đạt mức “chấp nhận được”.

### ứng dụng quadtree trong game

Quadtree thường được ứng dụng vào việc xét va chạm giữa các đối tượng, update và render (vẽ) các đối tượng.

Chi phí cho việc khởi tạo quadtree nhỏ hơn rất nhiều so với việc update, render, xử lý va chạm, … các đối tượng không cần thiết. Ví dụ một đối tượng nằm ngoài vùng nhìn thấy của camera trong game thì không nhất thiết phải render nó.

Khi sử dụng Quadtree, có thể dễ dàng nhận biết các đối tượng thuộc 2 node khác nhau không có khả năng va chạm với nhau (trừ một số trường hợp đặc biệt). Chi phí giảm đồng nghĩa với việc thời gian của một vòng lặp game được rút ngắn. Do đó FPS của game sẽ được cải thiện.

## CÁC KHÁI NIỆM LIÊN QUAN.

### Tổng quan về Mã nguồn mở.

#### Giới thiệu mã nguồn mở.

Trong lĩnh vực sản xuất và phát triển, mã nguồn mở là một triết lý hoặc một phương pháp thúc đẩy tái phân phối sản phẩm tự do và truy cập tới những thiết kế và ý tưởng của sản phẩm hoặc những chi tiết triển khai sản phẩm.

Những đặc trưng của mã nguồn mở:

* Mã nguồn mở là miễn phí
* Mã nguồn mở có thể truy cập được mã nguồn
* Mã nguồn mở phát triển theo thời gian
* Mã nguồn mở thì không đóng
* Mã nguồn mở có thể sử dụng lại và thay đổi
* Mã nguồn mở có thể triển khai ở bất kỳ nơi đâu và dành cho bất kỳ ai

#### So sánh mã nguồn mở và mã nguồn đóng

Với mã nguồn mở chúng ta có thể:

* Xem hoặc sửa đổi mã nguồn của phần mềm hoặc ứng dụng
* Phần mềm mã nguồn mở được phát hành bởi cộng đồng phát triển mã nguồn mở và phải trải qua giai đoạn thứ hai của quá trình phát triển những phần mềm mã nguồn mở còn phần mềm mã nguồn đóng thì được phát triển cô lập trong một nhóm nhỏ các nhà phát triển.
* Hỗ trợ nhiều cho nhà phát triển và có cộng đồng phát triển lớn
* Mã nguồn mở an toàn hơn và các lỗi và các lỗ hổng thường được sửa lỗi

Mã nguồn mở không có nghĩa là truy cập tới mọi mã nguồn. Các điều khoản phân phối phần mềm mã nguồn mở phải tuân thủ các tiêu chuẩn sau đây:

* Tái phân phối tự do

Giấy phép mã nguồn mở không được phép hạn chế bên nào, bên bán hoặc bên mua các phần mềm, nó như một thành phần của một phân phối thành phần tổng hợp có chứa các chương trình từ các nguồn khác nhau. Giấy phép không được phép yêu cầu trả tiền bản quyền hoặc lệ phí khác để bán như vậy.

* Mã nguồn
* Các công việc được kế thừa

Giấy phép phải cho phép sửa đổi và kế thừa và phải cho phép chúng phân phối theo các điều khoản tương tự như giấy phép của phần mềm gốc.

* Tính toàn vẹn mã nguồn của tác giả

Giấy phép có thể hạn chế mã nguồn

* Không kỳ thị với người hoặc nhóm
* Không kì thị đối với lĩnh vực của Endeavor
* Phân phối giấy phép
* Giấy phép không phải riêng cho một sản phẩm
* Giấy phép không được hạn chế các phần mềm khác
* Giấy phép phải trung lập với công nghệ

#### Giới thiệu phần mềm mã nguồn mở

##### Khái niệm

Phần mềm nguồn mở là phần mềm với mã nguồn được công bố và sử dụng một giấy phép nguồn mở. Giấy phép này cho phép bất cứ ai cũng có thể nghiên cứu, thay đổi và cải tiến phần mềm và phân phối phần mềm ở dạng chưa thay đổi hoặc đã thay đổi.

Không phải phần mềm mã nguồn mở nào cũng miễn phí. Và cũng không phải phần mềm miễn phí nào cũng là phần mềm mã nguồn mở.

Phần mềm mã nguồn mở hiểu với nghĩa rộng là một khái niệm chung được sử dụng cho tất cả các phần mềm mà mã nguồn của nó được công bố rộng rãi công khai và cho phép mọi người tiếp tục phát triển phần mềm đó. Điều này không có nghĩa là chúng có thể được sao chép, sửa chữa thoải mái hay sử dụng vào mục đích nào cũng được. Mã nguồn mở được công bố dưới rất nhiều điều kiện khác nhau (license), một số trong đó cho phép phát triển, sử dụng và bán tùy ý miễn là giữ nguyên các dòng về nguồn gốc sản phẩm (BSDL), một số bắt buộc tất các sản phẩm làm ra từ đó cũng phải là open-source (GPL), một số khác đòi hỏi phải công bố trọn vẹn mã nguồn (Mozilla), một số khác không cho phép sử dụng vào mục đích thương mại (Sun Solaris Source Code License), một số khác lại không có ràng buộc gì đáng kể (public domain, MIT X11 license) v.v. Qua đó ta thấy khái niệm open source không thể chuẩn xác mà muốn nói đến tính pháp lý của việc sử dụng các phần mềm mã nguồn mở, chúng ta phải xem xét đến điều kiện sử dụng (license) cụ thể mà dưới đó chúng được công bố. Một điều kiện hay được áp dụng nhất là GPL: GNU General Public License (<http://www.fsf.org/licenses/gpl.html>) của tổ chức Free Software Foundation.

GPL license có 2 đặc điểm phân biệt, đó là:

* Tác giả gốc giữ bản quyền về phần mềm nhưng cho phép người dùng rất nhiều quyền khác, trong đó có quyền tìm hiểu, phát triển, công bố cũng như quyền khai thác thương mại sản phẩm.
* Tác giả sử dụng luật bản quyền để bảo đảm các quyền đó không bao giờ bị vi phạm đối với tất cả mọi người, trên mọi phần mềm có sử dụng mã nguồn của mình.

Đặc biệt điểm thứ 2 thường được gọi là hiệu ứng virus (viral effect) vì nó biến tất cả các phần mềm có dùng mã nguồn GPL cũng biến thành phần mềm GPL. Trên thực tế điều này có ý nghĩa: bất kỳ tác giả nào sử dụng dù chỉ 1 phần rất nhỏ mã nguồn GPL trong chương trình của mình cũng phải công bố chương trình đó dưới điều kiện GPL.

Điều kiện này quy định ví dụ:

* Mọi phần mềm GPL đều phải công bỗ mã nguồn của mình rộng rãi công khai và phải tạo điều kiện cho mọi người truy cập được mã nguồn ấy (ví dụ qua web hoặc qua việc bán CD giá rẻ)
* Giữ nguyên mọi dòng chú thích về nguồn gốc tác giả, bản quyền của họ cũng như điều kiện được áp dụng đối với phần mềm (trong 1 file có tên LICENSE)
* Cấm việc bán mã nguồn nhưng cho phép kinh doanh chương trình được tạo ra từ mã nguồn ấy hoặc là các dịch vụ hỗ trợ liên quan.

##### Lợi ích của phần mềm mã nguồn mở

Lợi ích phần mềm nguồn mở thể hiện rỏ nhất ở tính kinh tế, sử dụng phần mềm nguồn mở tiết kiệm được nguồn tiền khổng lồ, nguồn tiền tiết kiệm trên sẽ giúp các nước đang phát triển hạn chế được hiện tượng chảy máu chất xám, khi mà các sinh viên được đào tạo về khoa học máy tính và phần mềm không còn đi tìm những công việc phù hợp với khả năng của họ tại các nước khác mà có thể làm việc tại đất nước mình.

Ở vấn đề giáo dục đào tạo phần mềm nguồn mở là là nền tảng cho việc giáo dục về khoa học máy tính, nếu dạy học về phần mềm sở hữu độc quyền, thì người học biết sẽ biết cách sử dụng phần mềm đó, nhưng nếu dạy và học về phần mềm nguồn mở thì người học không những biết cách sử dụng phần mềm nguồn mở mà còn biết thêm thông tin hoạt động của của phần mềm đó như thế nào. Song đôi lúc người ta lựa chọn phần mềm không chỉ dựa vào tính kinh phí phần mềm đó mà còn dựa vào độ chất lượng và ứng dụng của nó. Xét về phần mềm nguồn mở nó có các đặc điểm sau đây: tính an toàn, tính ổn định và đáng tinh cậy, giảm lệ thuộc vào xuất khẩu, vấn đề vi phạm bản quyền, quyền sở hữu trí tuệ và tính tuân thủ WTO, bản địa hóa, các chuẩn mở và sự không lệ thuộc vào nhà cung cấp, phát triển năng lực ngành công nghiệp địa phương.

Ở phần mềm nguồn mở hầu như không có Virus gây hại cho máy tính, đây cũng là vấn đề khiến mã nguồn mở ngày được quan tâm hơn so với phần mềm sử dụng mã đóng như Windows ví dụ như khi mua máy cài bản quyền Windows thì phải mua thêm phần mềm diệt Virus lại tiếp tục tốn tiền mua bản quyền phần mềm này.

Những ưu điểm phần mềm nguồn mở nói trên thể hiện như sau:

**Tính an toàn**

Mã nguồn được phổ biến rộng rãi: việc mã nguồn được phổ biến rộng rãi giúp người lập trình và người sử dụng dễ phát hiện, khắc phục các lỗ hỏng an toàn trước khi chúng bị lợi dụng. Đa phần các lỗi hệ thống của phần mềm nguồn mở được phát hiện trong quá trình rà soát định kỳ và được sửa trước khi gây ra bất kỳ thiệt hại nào. Các hệ thống phần mềm nguồn mở thường có quy trình rà soát chủ động chứ không phải rà soát đối phó.

Ưu tiên về tính an toàn đặt trên tiêu chí tiện dụng: có thể nói phần mềm nguồn mở được dùng để điều hành một phần lớn mạng internet và do đó nó nhấn mạnh nhiều đến tính bền vững, chức năng vận hành thay vì tính dễ sử dụng. Trước khi thêm bất cứ tính năng nào vào một ứng dụng phần mềm nguồn mở, bao giờ người ta cũng cân nhắc đến khía cạnh an toàn, và tính năng đó sẽ chỉ được đưa vào nếu không làm yếu đi tính an toàn của hệ thống.

Các hệ thống phần mềm nguồn mở chủ yếu dựa trên mô hình của Unix: nhiều người sử dụng, thuận tiện cho kết nối mạng. Do đó, chúng được thiết kế với một cấu trúc an toàn bảo mật cao. Điều này là đặc biệt quan trọng khi có nhiều người cùng chia sẻ quyền sử dụng một máy chủ cấu hình mạnh, bởi vì nếu hệ thống có độ an toàn thấp, một người sử dụng bất kỳ có thể đột nhập vào máy chủ, đánh cắp dữ liệu cá nhân của người khác, hoặc làm cho mọi người không tiếp cận được với các dịch vụ do hệ thống cung cấp. Kết quả của mô hình thiết kế này là chỉ có rất ít vụ tấn công được thực hiện thành công với các phần mềm nguồn mở.

Vậy tóm lại một gói phần mềm được tạo ra bởi một vài nhà thiết kế, hay một gói phần mềm do hàng nghìn nhà thiết kế sáng tạo nên người sử dụng sẽ chọn lựa như thế nào. Do phần mềm mã nguồn mở được sáng tạo bởi vô số các nhà thiết kế và người sử dụng nên độ bảo mật của chúng sẽ được cải thiện, cũng như chúng cũng sẽ được mang thêm nhiều tính năng mới và những cải tiến mới nên phần mềm mã mở sẽ dễ chú ý sử dụng hơn.

**Tính ổn định và đáng tin cậy**

Các phần mềm nguồn mở thường ổn định và đáng tin cậy đó là kết luận từ những cuộc phân tích, đánh giá và so sánh với các phần mềm nguồn đóng khác. Ví du như: một cuộc thử nghiệm theo phương pháp chọn ngẫu nhiên được tiến hành vào năm 1995, tập trung thử nghiệm 7 hệ điều hành thương mại và GNU/Linux. Người ta nạp vào các hệ điều hành này những tính năng ngẫu nhiên theo một trình tự lộn xộn, bắt chước hành động của những người sử dụng kém hiểu biết. Kết quả là các hệ điều hành thương mại có tỷ lệ xung đột hệ thống trung bình là 23% trong khi Linux chỉ bị lỗi vận hành trong 9% số lần thử nghiệm. Các tiện ích của GNU (phần mềm do FSF xây dựng trong khuôn khổ dự án GNU) bị lỗi vận hành có 6% số lần thử nghiệm. Nhiều năm sau, một nghiên cứu tiếp nối còn cho thấy tất cả những lỗi gặp trong cuộc thử nghiệm nói trên đều đã được khắc phục với hệ điều hành FOSS (FOSS là một thuật ngữ bao gồm bao gồm cả phần mềm tự do và phần mềm nguồn mở), trong khi với các phần mềm đóng thì vẫn hầu như chưa được đụng đến.

**Giảm lệ thuộc vào xuất khẩu**

Một trong những động cơ quan trọng khiến các quốc gia đang phát triển nhiệt tình hưởng ứng phần mềm nguồn mở chính là chi phí khổng lồ của giấy phép sử dụng các phần mềm đóng. Vì hầu như toàn bộ phần mềm của các nước đang phát triển đều được nhập khẩu, tiền mua những phần mềm này sẽ làm tiêu hao quỹ dự trữ ngoại tệ hết sức quý báu mà lẽ ra có thể được sử dụng hiệu quả hơn cho những mục tiêu phát triển khác. Công trình phần mềm nguồn mở tự do: nghiên cứu và khảo sát còn cho biết mô hình phần mềm nguồn mở này thiên nhiều hơn về dịch vụ công, do đó chi phí cho phần mềm cũng là để phục vụ những hoạt động của cơ quan Chính phủ chứ không phải cho mục đích lợi nhuận của các công ty đa quốc gia. Điều này có ảnh hưởng tích cực đến tạo công ăn việc làm cho xã hội, mở rộng năng lực đầu tư nội địa, và tăng thu cho ngân sách địa phương…

**Vấn đề vi phạm bản quyền, quyền sở hữu trí tuệ, và tính tuân thủ WTO**

Nạn sao chép phần mềm là vấn đề mà hầu như quốc gia nào trên thế giới cũng gặp phải. Tổ chức Business Software Alliance ước tính riêng trong năm 2002, tệ nạn này làm nước Mỹ thiệt hại mất 13.08 tỷ đô-la. Ngay với các quốc gia phát triển, nơi mà trên lý thuyết giá phần mềm còn vừa túi tiền người dân, tỷ lệ sao chép phần mềm vẫn ở mức rất cao (24% ở Mỹ và 35% ở Châu Âu). Tại các quốc gia đang phát triển, nơi mà mức thu nhập thấp khiến cho phần mềm trở thành một thứ hàng xa xỉ, thì tỷ lệ sao chép có thể đạt tới 90%. Nạn sao chép phần mềm và hệ thống luật pháp lỏng lẻo sẽ gây thiệt hại cho một quốc gia trên nhiều phương diện. Quốc gia nào yếu trong việc thực thi bảo vệ quyền sở hữu trí tuệ sẽ kém hấp dẫn với các nhà đầu tư nước ngoài. Quyền gia nhập WTO và khả năng tiếp cận những lợi ích mà tổ chức này mang lại bị ảnh hưởng khá nhiều bởi mức độ bảo vệ quyền sở hữu trí tuệ mà một quốc gia đạt được. Nạn sao chép phần mềm còn gây hại cho nền công nghiệp phần mềm nội địa, do các nhà lập trình địa phương giờ đây chẳng còn mấy động cơ để xây dựng những phần mềm bản địa.

**Bản địa hóa**

“Bản địa hoá là thích ứng một sản phẩm, làm cho nó phù hợp về mặt ngôn ngữ và văn hoá với thị trường mục tiêu (quốc gia hoặc địa phương), nơi sản phẩm được tiêu thụ và sử dụng”. Bản địa hoá là một trong những lĩnh vực nơi phần mềm nguồn mở tỏ rõ ưu thế của mình. Người sử dụng phần mềm nguồn mở có thể tự do sửa đổi để phần mềm trở nên thích ứng với những nhu cầu riêng biệt của một khu vực văn hoá đặc thù, bất kể quy mô kinh tế của khu vực đó. Chỉ cần một nhóm nhỏ những người có đủ trình độ kỹ thuật là đã có thể tạo ra một phiên bản nội địa ở mức độ thấp cho bất kỳ phần mềm nguồn mở nào. Còn việc xây dựng một hệ điều hành đã bản địa hóa hoàn chỉnh, mặc dù không đơn giản, nhưng ít ra cũng là khả thi. Việc Microsoft vào năm 1998 quyết định không xây dựng phiên bản Window 98 cho Iceland có thể đã gây nên những tác hại khó lường nếu như không có giải pháp thay thế của phần mềm nguồn mở. (Windows 98 là cải tiến của phiên bản trước, nó khá giống Windows 95. Một số cải tiến hữu ích như hỗ trợ USB, chia sẻ kết nối Internet)

**Các chuẩn mở và việc không lệ thuộc vào nhà cung cấp**

Sẵn có mã nguồn: với mã nguồn được phổ biến công khai, người ta lúc nào cũng có thể tái thiết kế và tích hợp lại bộ chuẩn của một ứng dụng. Mọi khả năng tuỳ biến đều đã thể hiện rõ trong mã nguồn, khiến cho không ai có thể giấu một chuẩn riêng trong một hệ thống phần mềm nguồn mở. Đối với phần mềm đóng thì việc tái thiết kế sẽ khó hơn. Một số mã còn được viết ra để đánh lạc hướng người dùng.

Chủ động tương thích chuẩn: khi đã có những chuẩn được thừa nhận rộng rãi, ví dụ như HyperText Markup Language (HTML) bộ chuẩn quy định cách thức hiển thị các trang web, thì các dự án phần mềm nguồn mở luôn chủ động bám sát những chuẩn này. Khi sử dụng các hệ thống phần mềm nguồn mở để thoát khỏi việc lệ thuộc vào nhà cung cấp.

Ví dụ: các doanh nghiệp có thể biến đổi một phần của gói phần mềm mã nguồn mở để biến chúng phù hợp với những nhu cầu của mình. Nhờ vào tính mở của các mã nguồn mà người sử dụng chỉ cần thay đổi mã nguồn để đạt được tính năng như ý muốn. Họ không thể làm được điều đó với các phần mềm có bản quyền.

##### Hạn chế của phần mềm mã nguồn mở

**Tính đa dạng và phức tạp**

Cộng đồng mã nguồn mở đã phát triển nhiều ứng dụng đa dạng với những chức năng tương tự nhau. Điều này gây khó khăn cho những người mới sử dụng trong việc chọn lựa. Cơ cấu chọn lựa đã được thiết lập như nhà sản xuất, giá cả, thị phần hoặc hỗ trợ chỉ cung cấp một sự giúp đỡ có hạn. Vấn đề thực sự là một khi gia tăng tính đa dạng sẽ dẫn đến sự phức tạp trong khi với xã hội ngày nay, người ta luôn mong muốn sự đơn giản.

**Sự dư thừa**

Sự chia nhánh mã nguồn có thể dẫn đến sự lãng phí trong quá trình phát triển nó. Nếu các nguồn phát triển được kết hợp và tổ chức lại một cách tốt hơn thì hiệu suất sẽ được nâng cao.

**Sự bất tiện**

Mã nguồn mở thường chỉ tập trung vào các mã của nó mà ít chú ý đến thiết kế giao diện và phát triển các tiện ích. Trong Microsoft World, hầu hết các phát triển trong vài năm gần đây đều thuộc lĩnh vực tiện ích và phát triển giao diện người dùng. Thêm vào những mâu thuẫn trên, người ta phải xem xét việc dùng một sản phẩm độc quyền chẳng hạn như của Microsoft vốn rất nổi tiếng với việc chuyển sang dùng phần mềm nguồn mở phải học cách sử dụng các ứng dụng mới.

**Thiếu các ứng dụng kinh doang đặc thù**

Mặc dù có rất nhiều dự án phần mềm nguồn mở đang được tiến hành, vẫn còn nhiều lĩnh vực hoạt động chưa có được một sản phẩm phần mềm hoàn thiện, đặc biệt là trong kinh doanh. Gần đây, sự ra đời của một số phần mềm quản lý nguồn lực của doanh nghiệp (Enterprise Resource Planning) của SAP (SAP được đánh giá là một trong bốn công ty software lớn nhất thế giới, sau Microsoft, IBM, và PeopleSoft), (PeopleSoft là nhà cung cấp phần mềm đứng hàng thứ 2 thế giới, sau hãng SAP của Đức, trong lĩnh vực phần mềm quản lý tài chính, nhân lực và kế toán) đã giúp đáp ứng phần nào nhu cầu của thị trường cao cấp, nhưng thị trường dành cho các doanh nghiệp vừa và nhỏ thì hầu như vẫn bị bỏ trống. Những phần mềm kế toán cơ bản, tiện lợi cho người dùng như Quickbooks, Peachtree hay Great Plains cho đến nay vẫn chưa có các phiên bản phần mềm nguồn mở tương đương.

**Tính tương hỗ với các hệ thống phần mềm**

Các phần mềm nguồn mở, nhất là khi cài trên máy để bàn, thường không hoàn toàn tương thích với phần mềm đóng. Với những tổ chức đã đầu tư nhiều cho việc thiết lập các định dạng lưu trữ dữ liệu và ứng dụng phần mềm đóng, việc cố gắng tích hợp những giải pháp phần mềm nguồn mở có thể sẽ rất tốn kém. Thay đổi các chuẩn đóng đã được xây dựng với mục đích ngăn chặn tích hợp những giải pháp thay thế sẽ chỉ càng làm trầm trọng thêm vấn đề.

**Nhược điểm khi đưa phần mềm mã nguồn mở thay thế phần mềm mã nguồn đóng**

Nhược điểm thứ nhất là do gặp sự kháng cự của người dùng do: không muốn thay đổi thói quen, phải học thêm một cái mới. Nhược điểm thứ hai: các công ty, cán bộ kỹ thuật chuyên về phần mềm nguồn mở ở Việt Nam hiện nay còn quá ít.

##### Các loại phần mềm mã nguồn mở

Phần mềm nguồn mở là những phần mềm được cung cấp dưới cả dạng mã và nguồn, không chỉ là miễn phí về giá mua mà chủ yếu là miễn phí về bản quyền: người dùng có quyền sửa đổi, cải tiến, phát triển, nâng cấp theo một số nguyên tắc chung qui định trong giấy phép phần mềm nguồn mở, chẳng hạn General Public Licence - viết tắt là GPL mà không cần xin phép ai, điều mà họ không được phép làm đối với các phần mềm nguồn đóng (tức là phần mềm thương mại).

Nhà cung cấp phần mềm nguồn mở có quyền yêu cầu người dùng trả một số chi phí về các dịch vụ bảo hành, huấn luyện, nâng cấp, tư vấn, vv... tức là những dịch vụ thực sự đã thực hiện để phục vụ người dùng, nhưng không được bán các sản phẩm nguồn mở vì nó là tài sản của trí tuệ chung, không phải là tài sản riêng của một nhà cung cấp nào.

Trên thị trường phần mềm, có nhiều loại giấy phép. Có thể chia các giấy phép này đại khái như sau:

**Phần mềm thương mại (Commercial Software).**

Là phần mềm thuộc bản quyền của tác giả hoặc nhà sản xuất, chỉ được cung cấp ở dạng mã nhị phân, người dùng phải mua và không có quyền phân phối lại.

**Phần mềm thử nghiệm giới hạn (Limited Trial Software).**

Là những phiên bản giới hạn của các phần mềm thương mại được cung cấp miễn phí nhằm mục đích thử nghiệm, giới thiệu sản phẩm và kích thích người dùng quyết định mua. Loại sản phấm này không chỉ giới hạn về tính năng mà còn giới hạn về thời gian dùng thử (thường là 60 ngày).

**Phần mềm “chia sẻ” (Shareware).**

Loại phần mềm này có đủ các tính năng và được phân phối tự do, nhưng có một giấy phép khuyến cáo các cá nhân hoặc tổ chức mua, tùy tình hình cụ thể. Nhiều tiện ích Internet (như “WinZip” dùng các thuận lợi của Shareware như một hệ thống phân phối).

**Phần mềm sử dụng phi thương mại (Non-commercial Use).**

Loại phần mềm này được sử dụng tự do và có thể phân phối lại bởi các tổ chức phi lợi nhuận. Nhưng các tổ chức kinh tế, thí dụ các doanh nghiệp ... muốn dùng phải mua. Netscape Navigator là một thí dụ của loại phần mềm này.

**Phần mềm không phải trả phần trăm cho nhà sản xuất (Royalties Free Binaries Software).**

Phần mềm được cung cấp dưới dạng nhị phân và được dùng tự do. Ví dụ: Bản nhị phân của các phần mềm Internet Explorer và NetMeeting.

**Thư viện phần mềm không phải trả phần trăm (Royalties Free Software Libraries).**

Là những phần mềm mà mã nhị phân cũng như mã nguồn được dùng và phân phối tự do, nhưng người dùng không được phép sửa đổi. Ví dụ: các thư viện lớp học, các tệp “header”.

**Phần mềm mã nguồn mở kiểu BSD - (Open Source BSD-style).**

Một nhóm nhỏ khép kín (closed team) đã phát triển các PMNM theo giấy phép phân phối Berkely (BSD – Berkely Software Distribution) cho phép sử dụng và phân phối lại các phần mềm này dưới dạng mã nhị phân và mã nguồn. Tuy người dùng có quyền sửa đổi mã, nhưng về nguyên tắc nhóm phát triển không cho phép người dùng tự do lấy mã nguồn từ kho mã ra sửa (gọi là check-out) và đưa mã đã sửa vào lại kho mã mà không được họ kiểm tra trước (gọi là các “check-in”).

**Phần mềm mã mở kiểu Apache (Open Source Apache-style).**

Chấp nhận nguồn mở kiểu BSD nhưng cho phép những người ngoài nhóm phát triển xâm nhập vào lõi của mã nền (core codebase), tức là được phép thực hiện các “check-in”.

**Phần mềm mã mở kiểu CopyLeft hay kiểu Linux (Open Source CopyLeft, Linux- style).**

Phần mềm mã mở kiểu CopyLeft (trò chơi chữ của Free Software Foundation – FSF – và GNU – Gnu’s Not Unix, để đối nghịch hoàn toàn với CopyRight!) hay còn gọi là giấy phép GPL (General Public Licence) là một bước tiến quan trọng theo hướng tự do hóa của các giấy phép phần mềm. Giấy phép GPL yêu cầu không những mã nguồn gốc phải được phân phối theo các qui định của GPL mà mọi sản phẩm dẫn xuất cũng phải tuân thủ GPL.

GPL cho người dùng tối đa quyền hạn và tự do đối với các phần mềm mã mở theo GPL, cụ thể người dùng có quyền không những sao chép, sửa đổi, mua bán các phần mềm mã mở dưới CopyLeft mà còn được quyền tự do như vậy đối với các phần mềm dẫn xuất. Tóm lại nếu phần mềm mã mở gốc đã theo CopyLeft thì mọi phần mềm mã mở dẫn xuất của nó cũng đương nhiên theo CopyLeft.

#### Giấy phép mã nguồn mở

##### Định nghĩa

Là giấy phép bản quyền dành cho các phần mềm máy tính trong đó đặt ra các quy định buộc người sử dụng phần mềm đó phải tuân theo. Giấy phép mã nguồn mở được sử dụng cho các phần mềm mã nguồn mở.

The Open Source Definition là định nghĩa được đề xuất bởi the Open Source Initiative, được sử dụng để mô tả có giấy phép đủ điều kiện là giấy phép "mã nguồn mở".

The Open Source Initiative (OSI) cũng xác nhận giấy để chứng nhận rằng họ thuộc The Open Source Definition.

Những nguyên tắc cơ bản của bản quyền mã nguồn mở:

Giấy phép mã nguồn mở bắt buộc không độc quyền khai thác thương mại các tác phẩm được cấp phép

Mã nguồn mở phải là mã nguồn đã được viết sẵn cho các công việc.

Mã nguồn mở phải cho phép việc tạo ra các sản phẩm đã được chỉnh sửa cho riêng người lập trình.

Mã nguồn mở không chỉ có nghĩa là được tiếp cận mã nguồn. Các điều khoản cho việc phát hành phần mềm nguồn mở phải tuân thủ với các tiêu chí sau đây:

* Miễn phí phát hành lại: Giấy phép không được yêu cầu trả tiền bản quyền, lệ phí khác để bán.
* Mã nguồn: Chương trình phải bao gồm mã nguồn, và phải có giấy phép phát hành trong mã nguồn.
* Nguồn gốc tác phẩm: Giấy phép phải cho phép sửa đổi và trích dẫn công việc, và phải cho phép họ được phát hành sản phẩm theo các điều khoản tương tự như giấy phép của phần mềm ban đầu.
* Tính toàn vẹn mã nguồn của tác giả: Giấy phép có thể hạn chế mã nguồn được phát hành mà không cho chỉnh sửa chỉ khi giấy phép cho phép phát hành "các tập tin bản vá" với mã nguồn cho các mục đích của việc sửa lỗi chương trình trong thời gian phát triển.
* Không phân biệt đối xử với người nào hoặc Nhóm nào: Giấy phép không được phân biệt đối xử chống lại bất kỳ người hoặc bất kỳ nhóm người nào.
* Không phân biệt đối xử với Lĩnh vực: Giấy phép không được hạn chế bất cứ ai sử dụng chương trình trong một lĩnh vực cụ thể nào.
* Phát hành của Giấy phép: Các quyền gắn liền với chương trình phải áp dụng cho tất cả các chương trình được phát hành lại mà không cần thiết phải thực hiện của một giấy phép bổ sung bởi các bên.
* Giấy phép không được cụ thể cho một sản phẩm: Các quyền gắn liền với chương trình không được phụ thuộc vào phần chương trình của một nhà phân phối phần mềm cụ thể.
* Giấy phép không được hạn chế các phần mềm khác: giấy phép không được đặt các hạn chế đối với các phần mềm khác được phân phối cùng với phần mềm được cấp phép.
* Giấy phép phải có công nghệ trung lập: Không có điều khoản của giấy phép được xác định dựa trên bất kỳ công nghệ cá nhân hoặc phong cách của giao diện

##### Tính pháp lý của giấy phép mã nguồn mở

Giấy phép mã nguồn mở vẫn duy trì xác nhận về bản quyền của tác giả gốc đối với phần mềm, tuy nhiên được đưa thêm các điều khoản để các hành vi phân phối, sửa đổi, sao chép… các phần mềm này trở thành hợp pháp. Vì vậy giấy phép này và các điều quy định trong nó có giá trị về mặt pháp lý (luật pháp được nhắc đến ở đây là một luật của Hoa Kỳ)

##### Phân loại giấy phép mã nguồn mở

Giấy phép mã nguồn mở có thể được chia thành 2 loại chính:

* Những giấy phép không quy định bất cứ sự hạn chế nào trong việc sử dụng mã nguồn (còn có thể gọi là các giấy phép không bảo hộ vì chúng không bảo vệ mã nguồn mở khỏi việc bị sử dụng trong các phần mềm không phải là mã nguồn mở)

Các giấy phép thuộc loại này: Apache Software License v.1.1, BSD License, Intel Open Source License for CDSA/CSSM Implementation, MIt License, Sun Industry Standards Source License, W3C Software Notice and License …

* Những giấy phép quy định các hạn chế trong việc sử dụng mã nguồn (còn có thê gọi là các giấy phép bảo hộ vì chúng đảm bảo rằng các mã nguồn mở khi được sử dụng trong bất cứ tình huống nào sẽ vẫn được công khai/miễn phí)

Các giấy phép thuộc loại này: Apple Public Source License v.1.2, Common Public License v.1.0, GNU General Public License v.2.0, IBM Public License v.1.0, Mozila Public License v.1.0 and v.1.1, Nokia Open Source License v.1.0a, Open Software License v.1.1, Python License, Python Software Foundation License v.2.1.1, Sun Public License v.1.0, …

Người giữ bản quyền mã nguồn khi sử dụng loại giấy phép không bảo hộ sẽ giữ lại bản quyền của họ đối với mã nguồn và cấp cho người được cấp bản quyền (có thể hiểu là người sử dụng sản phẩm, mã nguồn) tất cả các quyền thuộc về bản quyền của mã nguồn đó.

Người giữ bản quyền mã nguồn khi sử dụng loại giấy phép có bảo hộ giữ lại bản quyền của họ đối với mã nguồn và cấp cho người được cấp bản quyền tất cả các quyền thuộc về bản quyền của mã nguồn đó nhưng có ít nhất một điều kiện, thông thường là việc phân phối lại phần mềm/mã nguồn đó, dù đã được sửa đổi hay chưa, đều phải sử dụng cùng loại giấy phép ban đầu.

**Người viết giấy phép:**

Giấy phép mã nguồn mở do một số công ty, tổ chức lập ra để quy định về trách nhiệm của người sử dụng đối với một phần mềm/mã nguồn mở. Hiện tại, công ty, tổ chức OSI (Open Source Initiative) là người đưa ra định nghĩa về mã nguồn mở (OSD – Open Source Dèination) được cộng đồng công nhận rộng rãi. Các giấy phép mã nguồn mở đa phần được xây dựng dựa trên OSD.

**Quy trình thông qua một giấy phép mã nguồn mở tại OSI:**

* Cộng đồng thẩm định giấy phép sẽ thảo luận trong ít nhất 30 ngày
* Các ý kiến từ cộng đồng sẽ được tổng kết và đưa lên ban giám đốc OSI
* Ban giám đốc OSI sẽ đưa ra quyết định cuối cùng, hoặc yêu cầu các thông tin bổ sung, trong lần họp định kì tháng sau.
* Cộng đồng thẩm định sẽ được thông báo về quyết định của ban giám đốc OSI. Nếu giấy phép đó được chấp nhận, nó sẽ được đưa lên wedsite của OSI công nhận có thể xem tại: http://opensource.org/approval.

**Mục đích sử dụng:**

Các giấy phép mã nguồn mở được sử dụng để đảm bảo rằng các phần mềm, mã nguồn có sử dụng giấy phép này luôn là mã nguồn mở, phù hợp với OSD.

**Cách sử dụng giấy phép mã nguồn mở:**

Đối với nhà phát hành phần mềm, để có thể sử dụng một giấy phép mã nguồn mở có sẵn vào trong phần mềm của mình thì thông thường cần phải thực hiện các công việc sau:

* Đính kèm giấy phép vào trong phần mềm của mình (được hiểu là đưa nội dung bản giấy phép vào trong bộ cài đặt hoặc vào một file văn bản đi kèm với các file của chương trình)
* Điền các thông tin cân thiết vào trong giấy phép: mỗi giấy phép đều có hướng dẫn việc làm thể nào để sử dụng chúng, thông thường là điền tên tác giả, năm phát hành, công ty … vào trong các trường tương ứng được quy định sẵn của giấy phép.

#### Các loại giấy phép thông dụng

Tuy đa dạng về số lượng nhưng nhìn chung mỗi loại giấy phép mã nguồn mở đều gồm 3 nội dung chính: quyền lợi, trách nhiệm khi sử dụng, phân phối lại hay chỉnh sửa chương trình; xử lý nếu vi phạm giấy phép

##### Giấy phép GNU



Hình 2.14 Logo giấy phếp GNU

**Nhà phát hành**

Những giấy phép này áp đặt các quyền bị hạn chế từ những người tạo ra và phát hành các sản phẩm được viết lại dựa trên các sản phẩm sử dụng các giấy phép.

GNU General Public License (Giấy phép GPL) một cách rõ ràng đòi hỏi các sản phẩm được viết lại phải được phát hành cùng các điều khoản của GPLLicense và sản phẩm được viết lại chỉ được phép phát hành cùng các điều khoản của giấy phép.

Về cơ bản, việc sử dụng một sản phẩm được cấp phép GPL hoặc LGPL phải thỏa:

Giấy phép được cho phép không chỉ để tôn trọng bản quyền đối với sử dụng riêng các công việc được cấp phép

Tuy nhiên, để cho ra những hạn chế (và đối với GPL và LGPL giấy phép chỉ có những hạn chế) được cấp phép của bất kỳ sản phẩm được viết lại nào mà người đó chọn để tạo ra từ tác phẩm gốc.

GNU (General Public License)

Giấy phép cộng đồng của GNU hay GPL, là một trong những nền tảng của giấy phép mã nguồn mở.

Tạo bởi Free Software Foundation (FSF), đã có nhiều đóng góp không nhỏ cho mã nguồn mở, nó là giấy phép ưu tiên cho các dự án được ủy quyền bởi FSF, bao gồm Emacs GNU Editor và GNU C Compiler, trong số những điểm số của những người khác, bao gồm cả nhân GNU / Linux.

Bản quyền (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02.111-1.307, USA

Mọi người đều được phép sao chép và phát hành các bản sao đúng nguyên văn của tài liệu trong giấy phép này, nhưng không được phép thay đổi nó. Phiên bản 2 của giấy phép này được phát hành năm 1991, và phiên bản 3, phiên bản hiện tại phát hành năm 2007.

Nội dung giấy phép GNU được phát hành tại: [http://opensource.org/licenses/ gpl-license](http://opensource.org/licenses/%20gpl-license)

**Bản dịch tiếng việt của version 2.0 (GPLv2)**

Mở đầu:

Các giấy phép cho hầu hết các phần mềm được thiết kế tránh tự do chia sẻ và thay đổi nó. Ngược lại, GNU General Public License nhằm mục đích đảm bảo tự do chia sẻ và thay đổi phần mềm để đảm bảo rằng phần mềm này là miễn phí cho tất cả người lập trình của nó. Giấy phép Cộng đồng GNU này áp dụng cho hầu hết các phần mềm Free Software Foundation và bất kỳ chương trình khác mà tác giả của nó cam kết sử dụng giấy phép. (Thay vì vậy một số phần mềm Free Software Foundation khác được chấp nhận bởi Giấy phép GNU Library General Public License.). Người lập trình cũng có thể áp dụng nó cho các chương trình của người lập trình.

Khi chúng tôi nói về phần mềm miễn phí, chúng tôi đang đề cập đến tự do, không phải là giá. Giấy phép Công đồng của GNU được thiết kế để chắc chắn rằng người lập trình có quyền tự do phát hành bản sao của phần mềm miễn phí (và thu tiền cho dịch vụ này nếu người lập trình muốn), nhận mã nguồn hoặc có thể lấy nó nếu người lập trình muốn sử dụng mã nguồn đó, mà người lập trình có thể thay đổi phần mềm hoặc sử dụng đoạn nào của nó trong chương trình miễn phí mới; và người lập trình biết mình có thể làm những điều này.

Để bảo vệ quyền lợi của người lập trình, chúng tôi cần phải thực hiện hạn chế để cấm bất cứ ai phủ nhận những quyền này hay đòi hỏi từ bỏ các quyền. Những hạn chế trách nhiệm nhất định cho người lập trình nếu người lập trình phát hành bản sao của phần mềm, hoặc nếu người lập trình sửa đổi nó.

Ví dụ, nếu người lập trình phát hành bản sao của một chương trình, dù là miễn phí hay thu tiền, người lập trình phải trao cho người nhận tất cả các quyền mà người lập trình có. Người lập trình phải chắc chắn rằng họ cũng nhận được hoặc có thể nhận được mã nguồn. Và người lập trình phải cho người nhận biết những điều kiện này, để họ biết những quyền của họ.

Chúng tôi bảo vệ quyền của người lập trình với hai bước: (1) bản quyền phần mềm, và (2) cung cấp cho người lập trình giấy phép này cho phép người lập trình có quyền hợp pháp để sao chép, phát hành và / hoặc sửa đổi phần mềm.

Ngoài ra, để bảo vệ tác giả và chính chúng tôi, chúng tôi muốn chắc chắn là mọi người hiểu rằng không có bảo hành phần mềm miễn phí. Nếu phần mềm bị sửa đổi bởi người khác và thông qua đánh dấu thời gian, chúng tôi muốn người nhận biết rằng những gì họ có không phải là bản gốc, do đó bất kỳ lỗi nào do người khác làm , sẽ không ảnh hưởng đến danh tiếng đến tác giả gốc.

Cuối cùng, bất kỳ chương trình miễn phí luôn bị đe dọa bởi bằng sáng chế phần mềm. Chúng tôi muốn tránh nguy cơ việc những người phát hành lại chương trình tự do cá nhân sẽ giành bằng sáng chế, có hiệu lực làm cho chương trình độc quyền. Để ngăn chặn điều này, chúng tôi đã làm cho nó rõ ràng rằng bất kỳ bằng sáng chế phải được cấp phép cho sử dụng miễn phí của tất cả mọi người, hoặc ít nhất là không cấp cho tất cả.

**Các thuật ngữ chính xác và điều kiện để sao chép, phân phối và sửa đổi.**

Phần giới thiệu này một cách rõ ràng và chính xác đặt ra ba mục đích chính của GPL.

Đầu tiên, và quan trọng nhất cho tới bây giờ, là để giữ cho phần mềm miễn phí, rằng nó có thể được phân phối và sửa đổi mà không có sự cho phép bổ sung của nhà bản quyền.

Mục đích thứ hai của GPL là đảm bảo rằng cấp phép lưu ý rằng phần mềm theo giấy phép được phân phối "giống" và không có bảo hành.

Mục đích thứ ba là phần mềm được cấp phép miễn phí bằng sáng chế hạn chế, đến mức bằng sáng chế áp dụng đối với các phần mềm được cấp phép, phải được cấp phép tại song song với mã.

ĐIỀU KHOẢN VÀ ĐIỀU KIỆN PHÂN PHỐI, COPYING, SỬA ĐỔI.

* Giấy phép này áp dụng cho bất kỳ chương trình hoặc các công việc khác, trong đó có một thông báo được chèn vào bởi người giữ tác quyền nói rằng nó có thể được phân phối theo các điều khoản của Giấy phép Công cộng. "Chương trình" dưới đây, đề cập đến bất kỳ chương trình, công việc, và "sản phẩm dựa trên Chương trình" có nghĩa là Chương trình hay bất kỳ sản phẩm nào bắt nguồn từ theo quy định của pháp luật quyền tác giả: đó là để nói, một tác phẩm chứa chương trình hoặc một phần của nó, hoặc đúng nguyên văn hoặc với sự sửa đổi và / hoặc dịch sang ngôn ngữ khác. (Dưới đây, việc dịch ngôn ngữ được bao gồm nhưng không giới hạn trong "chỉnh sửa".) Cấp Giấy phép được gọi là "người lập trình". Các hoạt động khác hơn so với sao chép, phân phối và sửa đổi không được bao trả bởi Giấy phép này, chúng nằm ngoài phạm vi của nó. Hành động chạy Chương trình không bị hạn chế, và sản lượng từ Chương trình được bao phủ chỉ khi nội dung của nó tạo thành một sản phẩm dựa trên Chương trình (độc lập được thực hiện bằng cách chạy chương trình). Đó đúng hay không phụ thuộc vào những gì Chương trình tạo ra.
* Bạn có thể sao chép và phân phối các bản sao đúng nguyên văn của mã nguồn của Chương trình như bạn nhận được nó, trong bất cứ phương tiện nào, miễn là bạn công bố rõ ràng và thích đáng trên mỗi bản sao một thông báo về bản quyền và từ chối bảo hành, giữ nguyên vẹn  
  Tất cả các thông báo liên quan đến Giấy phép này và sự vắng mặt của bất kỳ bảo hành và cung cấp cho những người nhận Chương trình khác một bản sao của Giấy phép này cùng với Chương trình. Bạn có thể thu một khoản phí đối với hành vật lý của chuyển giao một bản sao, và bạn có thể bảo vệ tùy chọn bảo hành cung cấp của bạn có thu phí. Bạn có thể chỉnh sửa bản sao của bạn hoặc các bản sao của Chương trình hoặc bất kỳ phần nào của nó, do đó tạo thành một sản phẩm dựa trên Chương trình, và sao chép và phân phối những sửa đổi hoặc làm việc theo các điều khoản của phần 1 nêu trên, miễn là bạn cũng đáp ứng tất cả các điều kiện:
* Bạn phải làm cho các tập tin đổi chứa thông báo dễ thấy nói rằng bạn đã thay đổi các tập tin và ngày tháng của bất kỳ một thay đổi nào.
* Bạn phải làm bất cứ công việc nào mà bạn phát hành hay xuất bản, toàn bộ hoặc một phần có chứa hay bắt nguồn từ Chương trình hay bất kỳ phần nào của nó, được cấp toàn bộ miễn phí cho tất cả các bên thứ ba theo các điều khoản của Giấy phép này .
* Nếu chương trình đã chỉnh sửa thường đọc lệnh tương tác trong khi chạy, bạn phải làm nó, khi bắt đầu chạy để sử dụng tương tác theo cách thông thường nhất, in ra hay hiển thị một thông báo gồm có thông báo quyền tác giả thích hợp và thông báo rằng có bảo hành(nếu không, nói rằng bạn cung cấp một bảo hành) và người sử dụng có thể phát hành lại chương trình theo những điều kiện này, và báo cho người dùng biết cách xem một bản sao của Giấy phép này. (Ngoại lệ: nếu Chương trình chính nó là tương tác nhưng không in ra lời loan báo như vậy, công việc của bạn dựa trên Chương trình không yêu cầu phải có thông báo)
* Bạn có thể sao chép và phát hành Chương trình (hoặc một sản phẩm dựa vào nó, dưới Phần 2) trong mã đối tượng hoặc hình thức thực thi theo các điều khoản của Phần 1 và 2 bên trên, miễn là bạn cũng làm một trong những điều sau đây:
* Kèm theo toàn bộ mã nguồn tương ứng mà máy có thể đọc được, phải được phân phối theo các điều khoản của Phần 1 và 2 bên trên, trên một phương tiện thường dùng để trao đổi phần mềm;
* Kèm theo đó một đề nghị bằng văn bản, hợp lệ trong ít nhất ba năm, để cung cấp cho bất kỳ bên thứ ba, cho một khoản phí không nhiều hơn so với chi phí của bạn để phân phối mã nguồn, một máy tính có thể đọc được hoàn thành bản sao của mã nguồn tương ứng, để được phân phối theo các điều khoản của Phần 1 và 2 bên trên, trên một phương tiện thường dùng để trao đổi phần mềm;
* Kèm theo đó với những thông tin mà bạn đã nhận được đề nghị phát hành mã nguồn tương ứng. (Tùy chọn này chỉ cho phép phân phối phi thương mại và chỉ khi bạn nhận được chương trình theo mã đối tượng hoặc hình thức thực thi với đề nghị như vậy, phù hợp với Phần phụ (b) trên). Các mã nguồn cho một tác phẩm có nghĩa là hình thức ưa thích của tác phẩm cho khi sửa đổi nó. Đối với một công việc thực thi, mã nguồn có nghĩa là tất cả các mã nguồn cho tất cả các mô-đun đã chứa, cộng với bất kỳ tập tin giao diện định nghĩa liên quan, cùng với tập lệnh được sử dụng để kiểm soát trình biên dịch và cài đặt thực thi. Tuy nhiên, như là một ngoại lệ đặc biệt là mã nguồn được phát hành không cần không bao gồm bất cứ điều gì được phát hành bình thường (trong hoặc nguồn hoặc dạng nhị phân) với các thành phần chính (biên dịch, nhân, và như vậy) của hệ điều hành thực thi chạy, trừ khi thành phần kèm theo cần thực thi.
* Bạn không thể sao chép, sửa đổi, cấp phép con, hoặc phân phối chương trình ngoại trừ như được quy định trong Giấy phép này. Bất kỳ ý định sao chép, sửa đổi, cấp giấy phép phụ hay phát hành Chương trình bãi bỏ, và sẽ tự động chấm dứt quyền của bạn dưới Giấy phép này.
* Bạn không bắt buộc phải chấp nhận Giấy phép này, kể từ khi bạn chưa ký vào đó. Tuy nhiên, không có gì khác cho phép bạn sửa đổi hoặc phân phối chương trình hoặc các sản phẩm phái sinh của nó. Các hành động này bị pháp luật cấm nếu bạn không chấp nhận Giấy phép này. Vì vậy, bằng cách sửa đổi hay phát hành Chương trình (hay sản phẩm dựa trên Chương trình), bạn cho biết bạn chấp nhận Giấy phép này để làm như vậy, và tất cả các điều khoản và điều kiện cho việc sao chép, phát hành hay sửa đổi Chương trình hoặc làm việc dựa trên nó.
* Mỗi khi bạn phát hành lại Chương trình (hay sản phẩm dựa trên Chương trình), người nhận sẽ tự động nhận được giấy phép từ người cấp gốc, để sao chép, phân phối hoặc sửa đổi Chương trình với các điều khoản và điều kiện.
* Nếu, như một hệ quả của một phán quyết của tòa án hay các cáo buộc vi phạm bản quyền hoặc vì bất kỳ lý do nào khác (không giới hạn trong vấn đề bằng sáng chế), điều kiện đối với bạn (cho dù định của toà án, thoả thuận hoặc cách khác nào) mà mâu thuẫn điều kiện của Giấy phép này, họ không miễn cho bạn các điều kiện của Giấy phép này. Nếu bạn không có thể phân phối như vậy là thỏa mãn đồng thời nghĩa vụ của bạn dưới Giấy phép này và bất kỳ giao ước thích hợp khác, thì hậu quả là bạn có thể không phân phối chương trình ở tất cả. Ví dụ, nếu một bằng sáng chế không cho phép tái phân phối của Chương trình miễn phí bản quyền bởi tất cả những người nhận bản sao trực tiếp hoặc gián tiếp thông qua bạn, thì cách duy nhất bạn có thể thỏa cả bằng đó và Giấy phép này sẽ được hoàn toàn không phân phối của Chương trình. Nó không phải là mục đích của phần này để xúi giục bạn vi phạm bất kỳ bằng sáng chế hoặc quyền sở hữu khác hoặc tranh luận về tính hợp lệ của lời yêu sách như vậy, phần này có mục đích duy nhất của việc bảo vệ sự toàn vẹn của hệ thống phân phối phần mềm miễn phí, được thực hiện bởi thực hành giấy phép công cộng. Nhiều người đã đóng góp một loạt các phần mềm được phân phối thông qua hệ thống đó, sự phụ thuộc vào ứng dụng nhất quán của hệ thống đó, nó là đến tác giả / người tặng quyết định nếu người đó sẵn sàng để phân phối phần mềm thông qua bất kỳ hệ thống khác và được cấp phép không thể áp đặt sự lựa chọn đó.
* Nếu việc phân phối và / hoặc sử dụng Chương trình bị hạn chế trong một số quốc gia, hoặc bởi bằng sáng chế hoặc bản quyền, bản quyền giữ ban đầu đã đặt Chương trình dưới Giấy phép này có thể thêm sự hạn chế phát hành địa lý riêng loại trừ những quốc gia, do đó, cho phép phát chỉ ở các nước không bị loại trừ như thế. Trong trường hợp như vậy, Giấy phép này hợp nhất sự hạn chế như bằng văn bản trong cơ thể của Giấy phép này.
* Tổ chức Phần mềm Tự do có thể bản chỉnh sửa và / hoặc phiên bản mới của Giấy phép Công cộng theo thời gian. Phiên bản mới như vậy sẽ tương tự như trong tinh thần của phiên bản hiện, nhưng có thể khác nhau về chi tiết để giải quyết vấn đề mới.
* Nếu bạn muốn kết hợp các phần của Chương trình vào chương trình tự do khác mà điều kiện khác nhau, viết cho tác giả để xin phép. Đối với phần mềm được cấp bản quyền bởi Free Software Foundation, viết thư cho Tổ chức Phần mềm Tự do, đôi khi chúng ta có những ngoại lệ. Quyết định của chúng tôi sẽ được hướng dẫn bởi hai mục tiêu bảo tồn trạng thái tự do của tất cả các dẫn xuất của phần mềm miễn phí của chúng tôi và thúc đẩy việc chia sẻ và tái sử dụng phần mềm nói chung.

KHÔNG BẢO ĐẢM

* VÌ CHƯƠNG TRÌNH ĐÃ ĐƯỢC CẤP PHÉP MIỄN PHÍ, KHÔNG CÓ BẢO HÀNH ĐỐI VỚI CHƯƠNG TRÌNH MỞ RỘNG DƯỚI SỰ CHO PHÉP CỦA PHÁP LUẬT. TRỪ KHI CÓ VĂN BẢN XÁC ĐỊNH KHÁC, NHỮNG NGƯỜI GIỮ BẢN QUYỀN VÀ / HOẶC CÁC BÊN CUNG CẤP CHƯƠNG TRÌNH “NHƯ LÀ” ("AS IS") KHÔNG BẢO ĐẢM, rõ hay ngụ ý, BAO GỒM NHƯNG KHÔNG GIỚI HẠN Ở CÁC BẢO ĐẢM NGỤ Ý KHẢ NĂNG BÁN VÀ SỰ PHÙ HỢP CHO MỘT MỤC ĐÍCH CỤ THỂ. TOÀN BỘ RỦI RO VỀ CHẤT LƯỢNG VÀ HIỆU SUẤT CỦA CHƯƠNG TRÌNH. CHƯƠNG TRÌNH CÓ KHIẾM KHUYẾT, BẠN PHẢI CHỊU CHI PHÍ SỬA CHỮA DỊCH VỤ HAY TỰ SỬA CHỮA.
* TRONG TRƯỜNG HỢP YÊU CẦU CỦA LUẬT PHÁP HOẶC THỎA THUẬN VIẾT, HOẶC BÊN KHÁC CÓ THỂ SỬA ĐỔI VÀ / HAY PHÁT HÀNH LẠI CHƯƠNG TRÌNH NHƯ ĐƯỢC PHÉP. GIẤY PHÉP KHÔNG CHỊU TRÁCH NHIỆM VỚI BẠN VỀ THIỆT HẠI, BAO GỒM BẤT KỲ THIỆT HẠI CHUNG, NGẪU NHIÊN HAY LỢI NHUẬN, PHÁT SINH BÊN NGOÀI VIỆC SỬ DỤNG HAY KHÔNG CÓ KHẢ NĂNG SỬ DỤNG CHƯƠNG TRÌNH (BAO GỒM NHƯNG KHÔNG GIỚI HẠN MẤT DỮ LIỆU HAY DỮ LIỆU BỊ LÀM CHO KHÔNG CHÍNH XÁC HAY MẤT MÁT KHÁC HOẶC BÊN THỨ BA HOẶC CHƯƠNG TRÌNH VẬN HÀNH ĐƯỢC VỚI CÁC CHƯƠNG TRÌNH KHÁC, NGAY CẢ KHI ĐÃ ĐƯỢC KHUYÊN VỀ KHẢ NĂNG CỦA TỔN HẠI ĐÓ.

KẾT THÚC ĐIỀU KIỆN

**Làm thế nào để áp dụng điều kiện cho các chương trình mới của bạn**

Nếu bạn phát triển một chương trình mới, và bạn muốn nó được phần đông công chúng sử dụng, cách tốt nhất để đạt được điều này là làm cho phần mềm miễn phí mà mọi người có thể thay đổi theo các điều khoản.

Để làm như vậy, đính kèm thông báo sau vào chương trình. An toàn nhất là đính chúng vào đầu mỗi tập tin mã nguồn để có hiệu quả nhất khi bảo hành, và mỗi tập tin nên có ít nhất "bản quyền" dòng và con trỏ được thông báo đầy đủ.

dòng cho biết tên của chương trình và ý tưởng ngắn gọn về những gì nó làm.

Bản quyền (C)

Chương trình này là phần mềm miễn phí, bạn có thể phát hành lại nó và / hoặc sửa đổi nó theo các điều khoản của GNU General Public License như được xuất bản bởi Free Software Foundation, phiên bản 2 của Giấy phép, hoặc (tùy chọn) phiên bản bất kì sau này.

Chương trình này được phân phối với hy vọng rằng nó sẽ hữu ích, nhưng KHÔNG CÓ BẢO HÀNH, không bảo đảm ngụ ý cho BÁN hoặc PHÙ HỢP CHO MỘT MỤC ĐÍCH CỤ THỂ. Xem GNU General Public License để biết thêm chi tiết.

Bạn cần một bản sao của GNU General Public License cùng với chương trình này, nếu không, hãy viết thư cho Free Software Foundation, Inc, 59 Nơi Temple, Suite 330, Boston, MA 02.111-1.307 USA

Được ký bởi Ty Coon, 1 Tháng Tư 1989 Ty Coon, chủ tịch của Vice

**Nội dung chính**

Quyền lợi:

* Quyền được sao chép, phân phối chương trình, quyền được yêu cầu trả phí cho việc phân phối đó
* Quyền được thay đổi chương trình để sử dụng cho mục đích cá nhân
* Quyền được phân phối bản đã được thay đổi đó

Nghĩa vụ:

* Khi sao chép và phân phối chương trình, phải đính kèm các thông báo về bản quyền gốc và không nhận bảo hành (trừ trường hợp có văn bản thêm về quy định bảo hành.)
* Khi phân phối bản đã được thay đổi bởi bản thân, phải chú thích rõ đó là bản đã được thay đổi, các thành phần được thay đổi và áp dụng giấy phép GNU cho bản đã được thay đổi đó.
* Khi phát hành chương trình phải công khai mã nguồn của chương trình của nó đồng thời phải công bố mã nguồn của chương trình tối thiểu 3 năm mà không được đòi một khoản chi phí nào từ những yêu cầu mã nguồn trừ chi phí vận chuyển hay tương đương.

Xử lý vi phạm:

Người vi phạm giấy phép bị tước quyền sử dụng giấy phép GNU, tuy nhiên nếu B thừa hưởng giấy phép từ A, mà A vi phạm mà B không vi phạm thì B vẫn giữ được giấy phép GNU.

Ta có thể thấy giấy phép GNU có những điểm đáng chú ý sau:

* Có đặc tính virus bởi sức lây lan và kế thừa của nó. Một người nhận sản phẩm từ người mang giấy phép GNU thì ngay lập tức người đó cũng mang giấy phép GNU. Hệ quả là GNU là giấy phép phổ biến nhất, bỏi mã nguồn luôn ở dạng công cộng cho phép ai cũng có thể tham gia ở bất kì thời điểm nào.
* Tác giả gốc giữ bản quyền và cho người dùng các quyền hợp pháp trong việc: sao chép, chỉnh sửa, phân phối sản phẩm. Mặc dù giấy phép GNU yêu cầu mã nguồn và chương trình phải được cung cấp miễn phí, song nó cho phép người phân phối có thể kinh doanh với sản phẩm nhờ đưa ra các chính sách về bảo hành, tính chi phí phân phối sản phẩm, đào tạo sử dụng, …

Một số người lo ngại vấn đề mã nguồn bị public tạo điều kiện cho hacker tìm ra lỗ hổng, song thực tế lại chứng minh ngược lại, càng cố che giấu mã nguồn thì lại càng bị tấn công nhiều hơn.

**Các phần mềm sử dụng giấy phép GNU GPL nổi bật:**

* RedHat Enterprise Linux
* Ubuntu
* GIMP
* Drupal, WordPress, Jooomla…

##### Giấy phép BSD

**Nhà phát hành**

Giấy phép BSD (Berkeley Software Distribute License) là một giấy phép phần mềm tự do với các điều kiện rất đơn giản được sử dụng rộng rãi cho phần mềm máy tính. Ban đầu giấy phép BSD được thiết kế bởi đại học California tại Berkeley năm 1980 cho dự án BSD

**Nội Dung Chính**

Quyền lợi:

Giấy phép BSD cho phép sử dụng và phân phối lại mã nguồn và sản phẩm có hoặc không có sửa đổi miễn là tuân thủ các yêu cầu sau:

Nghĩa vụ:

* Phải giữ nguyên thông báo bản quyền của sản phẩm. Yêu cầu này nhằm đảm bảo một người dùng bất kì không thể tuyên bố anh ta đã viết ra một phần mềm nếu thực sự anh ta không viết ra nó.
* Phải kèm theo 2 thông báo: Danh sách các điều kiện và từ chối trách nhiệm
* Không sử dụng tên dự án hay tên nhà phân phối vào mục đích quảng bá bản thân nếu không được cho phép.

Như vậy, có thể thấy đặc điểm lớn nhất của BSD là nó cho phép các nhà phát triển phần mềm có thể thương mại hóa một cách thực sự các sản phẩm phần mềm có sử dụng mã nguồn mở dùng giấy phép BSD, tức kiếm tiền dựa trên mã nguồn của chương trình (chủ yếu mã nguồn do họ viết thêm và giữ lại mã nguồn đã được sửa đổi đó cho bản thân mà không công bố) thay vì chỉ có thể kiếm tiền từ các hoạt động không trực tiếp gắn với phần mềm như bảo hành, phát hành, đào tạo hướng dẫn sử dụng… Ngoài ra, giấy phép BSD còn cho phép các nhà phát triển thay thế, bổ sung thêm các điều khoản vào trong giấy phép cho phù hợp với mình hoặc thậm chí sử dụng một giấy phép khác.

**Một số phần mềm sử dụng giấy phép BSD**

* Một số lớn các dự án xây dựng phần mềm nguồn mở, bao gồm cả những phần mềm lớn, đã được cấp phép dạng BSD, ví dụ:
* Hệ thống windows Xfree86: nền tảng của hầu hết các giao diện với người sử dụng trong các hệ thống phần mềm nguồn mở
* FreeBSD, NetBSD, OpenBSD: các biến thể của phiên bản Unix gốc được cấp giấy phép BSD, cả 3 đều sử dụng rộng rãi trên mạng Internet, đặc biệt là FreeBSD, chương trình điều hành Yahoo và dịch vụ Hotmail.
* Tích hợp mã nguồn được cấp phép theo chế độ giấy phép BSD vào các ứng dụng thương mại và việc khác đơn giản. Ngay Microsoft trước đây cũng đã từng sử dụng một số mã BSD trong phần kết nối mạng của mã nguồn Window. Nhiều công ty dưa cả phần mềm máy chủ apache vào trong gói phần mềm thương mại mà học cung cấp cho khách hàng.

##### Giấy phép MIT

**Nhà phát hành**

MIT là một giấy phép phần mềm tự do được phát hành bởi học viện công nghệ Massachusetts (Massachusetts Institute of Technology - MIT) được hội đồng MITX sử dụng.

Theo tổ chức phần mềm tự do, giấy phép MIT nên được gọi chính xác hơn là giấy phép X11, học viện Massachusetts đã từng dùng nhiều loại giấy giấy phép khác nữa, và do giấy phép này ban đầu soạn thảo cho X Window System (dự án tạo giao diện đồ hòa cho người dùng ở các máy tính kết nối mạng vào năm 1984).

**Nội dung chính**

MIT là một giấy phép dạng “permissive”, nghĩa là cho phép tái sử dụng các phần mềm độc quyền nhưng với điều kiện giấy phép MIT đã được phân phối kèm phần mềm đó. Giấy phép MIT cũng tương thích GPL, nghĩa là GPL cho phép tổng hợp và tái phân phối các phần mềm sử dụng giấy phép MIT

* Giấy phép MIT không bắt buộc phải công khai mã nguồn
* Người sử dụng phải kèm theo giấy phép MIT vào bản chỉnh sửa của mình, tuy nhiên không bị bắt buộc phải sử dụng giấy phép MIT cho toàn bộ bản đó
* Giấy phép MIT cũng có thể được chỉnh sửa lại cho phù hợp với thực tế

Ngoài ra, giấy phép MIT không cấm sử dụng tên của người giữa bản quyền vào mục đích quảng bá và cũng không bắt buộc phải hiện danh sách tất cả những người từng tham gia thực hiện dự án trong phần about của chương trình.

**Một số phần mềm sử dụng giấy phép MIT**

Các gói phần mềm sử dụng giấy phép MIT bao gồm Expat, Pu TTY, Ruby on Rails, Lua 5.0 và X Window System.

Cũng có một số sản phẩm sử dụng chung giấy phép MIT với một giấy phép khác, mà nổi tiếng nhất là thư viện JQuery của javascript, sản phẩm này sử dụng cả giấy phép MIT và GNU

Có thể xem danh sách chi tiết các sản phẩm sử dụng giấy phép MIT tại: <http://en.wikipedia.org/wiki/List_of_software_that_uses_the_MIT_license>

##### Giấy phép Apache

Description: ASF logo

Hình 2.15 Logo giấy phép Apache

**Nhà phát hành**

Giấy phép Apache là giấy phép mã nguồn mở được soạn ra bởi tổ chức phần mềm Apache (ASF – Apache Software Foundation)

Tất cả mọi phần mềm do ASF phát hành đều mang giấy phép Apache. Những dự án không thuộc ASF nhưng vẫn mang giấy phép Apache, cho đến tháng 7 năm 2009 là vào khoảng 500 dự án.

Phiên bản mới nhất của Apache là 2.0: <http://apache.org/licenses/LICENSE-2.0>

ASF và tổ chức phần mềm tự do (FSF) đều công nhận giấy phép Apache 2.0 là một giấy phép phần mềm tự do, tương thích với phiên bản giấy phép GNU 3.0

**Nội dung chính**

* Giống như các giấy phép mã nguồn mở khác, giấy phép Apache cho phép người dùng tự do sử dụng phần mềm với bất kì mục đích nào, tự do phân phối bản sửa đổi mình làm.
* Giấy phép Apache không yêu cầu bản sửa đổi của phần mềm phải được phân phối dưới cùng giấy phép với bản gốc, cũng không yêu cầu bản sửa đổi phải được phân phối dưới dạng mã nguồn mở. Giấy phép Apache đã được sử dụng trong sản phẩm họ nhận được.

Như vậy, trái với giấy phép copyleft, người nhận được những bản sửa đổi của chương trình mang giấy phép Apache cũng không nhất thiết phải nhận toàn bộ những quyền trên. Nói cách khác là họ có nhận được quyền sử dụng chương trình và mã nguồn theo cách họ muốn, kể cả việc giữ lại mã nguồn cho riêng mình

Có hai file cần được đặt trong thư mục gốc khi phân phối chương trình:

* LICENSE: bản copy của chính giấy phép MIT
* NOTICE: văn bản chú thích tên của các thư viện đã dùng , kèm tên người phát triển
* Trong mỗi tệp tin đã được cấp phép, bất kì thông tin về bản quyền và bằng sáng chế trong bản phân phối lại phải được giữ nguyên như bản gốc và ở mỗi tệp tin đã được chỉnh sửa phải thêm vào ghi chú là đã được chỉnh sửa khi nào

Giấy phép Apache không yêu cầu trích dẫn toàn bộ giấy phép vào sản phẩm hay tệp tin đính kèm bản phân phối, mà chỉ cần thêm phần thông báo có chứa đường link tới wedsite chứa giấy phép (GNU 3.0 cũng áp dụng cách này). Khi đó, người dùng chỉ thấy thông báo ngắn gọn như sau:

Copyright [yyyy] [name of copyright owner]  
Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>  
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and limitations under the License.

So với giấy phép BSD, Apache có nội dung chặt chẽ hơn trong các điều khoản cà duy trì quyên sở hữu trí tuệ (BSD chỉ có 3 yêu cầu ngắn gọn trong giấy phép, trong khi Apache có tới khoản 9 mục lớn).

**Các phần mềm sử dụng giấy phép Apache**

Mặc dù, nhiều phần mềm không phải do ASF phát hành nhưng cugnx mang giấy phép Apache. Song, nổi tiếng và sử dụng nhiều nhất vẫn là Apache Server, phần mềm giao tiếp dành cho máy chủ. Ngay từ khi ra đời, Apache đã có thể cạnh tranh với chương trình máy chủ của Nescape và tới nay thì Apache chiếm khoảng hơn 60% thị phần máy chủ thế giới.

Bên cạnh Apache Server, còn có những phần mềm sử dụng giấy phép Apache như:

* Apache Cocoon – một chương trình nền cho ứng dụng wed
* XAMPP – gói ứng dụng wed gồm Apache và MySQL
* Apache Axis2 – chương trình nền cho dịch vụ wed (xử lý được cả ngôn ngữ Java và C)

##### Giấy phép Artistic

**Nhà phát hành**

Giấy phép Artistic do Larry Wall thuộc tổ chức Perl (The Perl Foundation) viết, giấy phép Artistic được sử dụng chủ yếu cho các gói phần mềm miễn phí và mã mở, điển hình nhất là trong việc hoàn thiện ngôn ngữ Perl và các module của CPAN (gói chưa hơn 18.000 module phần mềm viết bằng ngôn ngữ Perl).

Tuy nhiên, việc xếp Artistic vào danh sách các giấy phép phần mềm lại từng gây ra nhiều tranh cãi. Tổ chức phần mềm tự do (FSF) đã chỉ trích giấy phép Artistic “quá mơ hồ, một số đoạn chỉ nhằm lợi ích của mình và ý nghĩa không rõ ràng”. FSF cho rằng giấy phép này không thể sử dụng nếu đứng đơn lẻ được, nhưng vẫn chấp nhận giấy phép Artistic trở thành một trong hai giấy phép chính cho các dự án Perl

Bản đầy đủ của giấy phép tại địa chỉ: <http://www.opensource.org/licenses/artistic-license-2.0.php>

**Nội dung, đặc điểm chính**

Giấy phép Artistic có nhiều điểm khác so với các giấy phép đã nêu ở trên, cụ thể:

* Quyền sử dụng và chỉnh sửa nhưng không được phân phối: người dùng được phép sử dụng bản đã sửa chữa vào bất kì mục đích nào, không giới hạn, miễn là không phát tán bản đã chỉnh sửa đó(khoản 1 của giấy phép)
* Quyền phân phối bản chỉnh sửa dưới dạng mã nguồn, nếu bạn đảm bảo ít nhất 1 trong 4 điều sau:

Người giữ bản quyền của bản gốc biết được bản sửa đổi, vẫn dưới giấy phép trước đó, người giữ bản quyền có thể thêm chỉnh sửa vào bản gốc (khoản mục 4a). Điều này đồng nghĩa với, người nắm quyền gốc đầu tiên có quyền quyết định nâng cấp sản phẩm của mình rộng rãi hay không

* Đảm bảo rằng việc cài đặt bản chỉnh sửa không ngăn cản người dùng cài đặt bản gốc của chương trình. Thêm vào đó, tên của bản chỉnh sửa phải khác với tên của bản gốc.
* Cho phép bất kì ai nhận phiên bản chỉnh sửa, có quyền phân phối mã nguồn của bản chỉnh sửa theo giấy phép Artistic hoặc một giấy phép tương đương (điều 4 cii). Điều này đảm bảo quyền lợi của mọi người như nhau.

Một vài điểm khác biệt của giấy phép Artistic:

Điểm khác biệt lớn trong giấy phép Artistic nằm ở điều (7) và điều (8) của giấy phép: tập hợp và kết nối các gói.

* Có thể tập hợp gói (gồm cả bản gốc và bản chỉnh sửa) với các gói khác và phân phối sản phẩm đã được tổng hợp này, miễn là đảm bảo không thu phí cấp giấy phép cho gói. Bạn được phép thu phí cho việc phân phối, ngay cả thu phí cho những thành phần khác trong gói. Có thể ví dụ: bạn download một phần mềm A về, sữa chữa, cải tiến nó thành phần mềm B, bạn có thể gói cả A,B với các phần mềm khác (cũng mang giấy phép này) thành một gói sản phẩm đồ sộ X và thu phí bản phân phối X này
* Cho phép nối các bản gốc hoặc bản chỉnh sửa của nhiều sản phẩm khác nhau. Đưa cả gói (như trên) vào dự án lớn hơn, hoặc tạo nên một chương trình duy nhất và phân phối chương trình đó hoàn toàn theo ý mình, miễn là sản phẩm kết quả không được ảnh hưởng tới các phần trong gói và giao diện không phải là giao diện của gói.

Ví dụ: Sau khi bạn có gói X như trên, bạn có thể phát triển tiếp X thành Y, Y này phải là ứng dụng độc lập, không ảnh hưởng tới việc sử dụng A và B và không mang giao diện của A và B.

**Các phần mềm sử dụng giấy phép**

Được sử dụng cho bộ mô phỏng SNEeSe và FakeNES, Paros Proxy, JavaFBP toolkit và NcFTP.

#### Vần đề tương thích của các giấy phép

Sau khi xem xét một số giấy phép ở trên, Các giấy phép đều nhằm mục tiêu đảm bảo cho người dùng có quyền sử dụng, sửa chữa, tái phân phối sản phẩm gốc hoặc phân phối sản phẩm mình đã sửa chữa. Cho dù các điều khoản có khác nhau, nhưng đã là giấy phép phần mềm tự do thì lúc nào cũng hướng tới những mục đích này.

Tuy vậy, vẫn có những điểm khá biệt giữa các giấy phép, như việc người dùng sau khi sửa đổi một chương trình mã nguồn mở, có được giữ chương trình trình đó lại cho riêng mình không? Việc một chương trình mã nguồn mở cần thông báo danh sách tất cả mọi người từng tham gia vào dự án, hay chỉ đề tên người đầu tiên khởi xướng dự án? Việc nên hay không đem đặc đặc tính “virus” của GNU vào tất cả các giấy phép khác? Việc sử dụng giấy phép “cứng” không cho phép sửa đổi hay dùng giấy phép “mềm” cho phép sửa đổi các điều khoản trong giấy phép?

Nhiều yêu cầu khiến những người soạn thảo giấy phép phải họp nhau lại và cố gắng tìm ra một cuẩn chung chô giấy phép mã nguồn mở, hoặc chí ít có thể tạo ra sự tương thích giữa các giấy phép.

Giấy phép GNU do tổ chức phần mềm tự do phát hành thường được chọn để so sánh với các giấy phép khác. Hiển nhiên, GNU cũng không phải là hoàn thiện hoàn toàn nếu đem so sánh về mặt thương mại, nhưng nó thực sự là giấy phép mở toàn diện.

Bảng so sánh một số giấy phép mã nguồn mở

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Giấy phép | Khả năng một phần mềm/thư viện mã nguồn đóng(A) liên kết với một phần mềm/thư viện mang giấy phép mở tương ứng | Khả năng kết hợp một phần mềm (A) với một phần mềm/thư viện mang giấy phép mở tương ứng | Tái phân phối mã nguồn sau khi thay đổi | Khả năng tương thích với GNU GPL |
| GPL | Không được bởi phần mềm (A) nếu liên kết thì sẽ bị coi như một sản phẩm phát sinh | Không được đối với phần mềm (A) có giấy phép không tương thích với GNU GPL | Chỉ khi nó (mã nguồn sau khi thay đổi) cũng mang giấy phép GNU | có |
| LGPL (Less GPL) | Có (bởi phần mềm sử dụng một thư viện không được coi là một sản phẩm phát sinh từ thư viện đó) | Được nhưng có vài giới hạn: phải cung cấp mã nguồn của thư viện LGPL đã phân phối kèm bản chỉnh sửa (nếu có), nếu cần sửa chữa thư viện LGPL thì nên có sự đồng ý trước | Chỉ khi nó (mã nguồn sau khi thay đổi) cũng mang giấy phép LGPLhhay GPL | có |
| Apache Public | Có | Có | Có (chừng nào tên Apache không được sử dụng trong tên của sản phẩm phát sinh) | Không |
| Artistic 2.0 | Có | Có (chừng nào chương trình cin viết bằng C hoặc Perl và được nhúng vào) | Cho phép nếu thõa mãn ít nhất một trong các điều kiện đã nói trong mục 5b | có |
| BSD | Có | Có | Có | Chỉ khi bản chỉnh sửa của giấy phép BSD tương thích với GPL. Bản BSD gốc không tương thích do có điều khoản về ngăn quảng cáo |
| MIT (X11) | Có | Có | Có | Có |
| Sun Public | Có | Có | Chỉ được phát sinh | Không |

Bảng 2.1 So sánh một số giấy phép mã nguồn mở

Bản so sánh đầy đủ hơn gồm 15 loại giấy phép có thê tham khảo tại địa chỉ: <http://developer.kde.org/documentation/licensing/licenses_summary.html>

#### Chọn giấy phép phù hợp

Hiện nay, hai giấy phép GNU và BSD theo thống kê là hai giấy phép mã nguồn mở sử dụng phổ biến nhất:

* Với những sản phẩm có dung lượng nhỏ, sử dụng đơn giản, nhà phát hành có thể sử dụng giấy phép BSD cho sản phẩm đó, vì BSD không bắt buộc phải công khai mã nguồn và hầu như không có người sử dụng nào muốn bỏ tiền ra để được bảo hành/đào tạo hướng dẫn sử dụng với một sản phẩm.
* Ngược lại, những sản phẩm lớn, phức tạp, nhà phát hành lúc này có thể quan tâm đến giấy phép GNU và thu lợi nhuận từ việc bảo hành, phân phối và đào tạo hướng dẫn sử dụng.

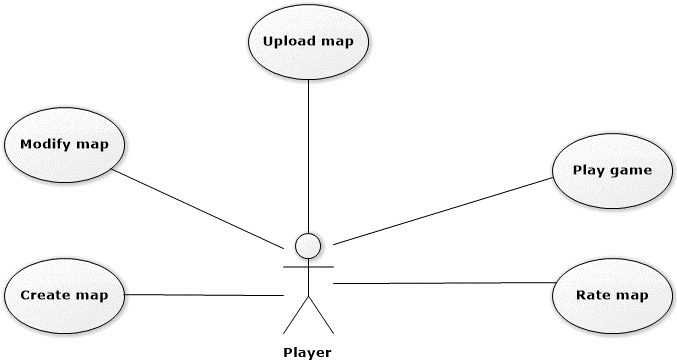
Nếu nhà phát hành không quan tâm đến lợi nhuận thì cả BSD và GNU đều tốt và phần lớn mã nguồn trên trang wed sourceforge.net mang giấy phép GNU

# Hiện thực hóa

## Client-side

### Các use case chính

#### Mô hình Use-case



Hình 3.1 Sơ đồ Use-case

* Danh sách Actor:

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Actor** | **Ý nghĩa/ghi chú** |
| 1 | Player | Người chơi game |

Bảng 3.1 Danh sách Actor

* Danh sách Use-case:

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Use-case** | **Ý nghĩa/ghi chú** |
| 1 | Create map | Người chơi tạo 1 map mới |
| 2 | Modify map | Người chơi chỉnh sửa map của mình |
| 3 | Upload map | Người chơi upload map lên server |
| 4 | Play game | Người chơi chơi game |
| 5 | Rate map | Người chơi đánh giá 1 map |

Bảng 3.2 Danh sách use-case

#### Đặc tả Use-case

* Create map:

|  |
| --- |
| * Tóm tắt: |
| * Tác nhân: ngươi chơi game. * Use-case này cho phép người chơi tạo thêm 1 map mới theo ý thích của mình |
| * Dòng sự kiện: |
| * Dòng sự kiện chính:  1. Người chơi bấm chọn Map editor trong main menu. 2. Người chơi bấm nút tạo map mới. 3. Người chơi thêm các đối tượng gạch, đá theo ý thích mình. 4. Người chơi nhấn nút ok để test thử map của mình. |
| * Các yêu cầu đặc biệt: |
| * Không có. |
| * Trạng thái hệ thống khi bắt đầu Use-case: |
| * Game đã được khởi động |
| * Trạng thái hệ thống khi kết thúc Use-case: |
| * Một scene mới được tạo ra để player test game |

Bảng 3.3 Đặc tả use-case create map

* Modify map:

|  |
| --- |
| * Tóm tắt: |
| * Tác nhân: ngươi chơi game. * Use-case này cho phép người chơi chỉnh sửa map đã tạo |
| * Dòng sự kiện: |
| * Dòng sự kiện chính:  1. Người chơi bấm chọn Map editor trong main menu. 2. Người chơi bấm nút chỉnh sửa map. 3. Người chơi thêm các đối tượng gạch, đá theo ý thích mình. 4. Người chơi nhấn nút ok để test thử map của mình.  * Dòng sự kiện khác:   Không có. |
| * Các yêu cầu đặc biệt: |
| * Không có. |
| * Trạng thái hệ thống khi bắt đầu Use-case: |
| * Game đã được khởi động |
| * Trạng thái hệ thống khi kết thúc Use-case: |
| * Một scene mới được tạo ra để player test game |

Bảng 3.4 Đặc tả use-case modify map

* Upload map:

|  |
| --- |
| * Tóm tắt: |
| * Tác nhân: ngươi chơi game * Use-case này cho phép người chơi upload map đã tạo lên server |
| * Dòng sự kiện: |
| * Dòng sự kiện chính:  1. Người chơi đã tạo map và đang test map 2. Người chơi test thành công (đã finish) 3. Người chơi nhấn nút upload  * Dòng sự kiện khác:   2.1: người chơi thất bại do tạo map quá khó |
| * Các yêu cầu đặc biệt: |
| * Thiết bị người chơi phải có kết nối internet |
| * Trạng thái hệ thống khi bắt đầu Use-case: |
| * Game đã được khởi động * Server đã được khởi động |
| * Trạng thái hệ thống khi kết thúc Use-case: |
| * Client hiển thị kết quả upload map * 1 map mới được lưu thêm vào database của server |

Bảng 3.5 Đặc tả use-case upload map

* Play game:

|  |
| --- |
| * Tóm tắt: |
| * Tác nhân: ngươi chơi game. * Use-case này cho phép người chơi chọn và chơi game theo map mình chọn |
| * Dòng sự kiện: |
| * Dòng sự kiện chính:  1. Người dùng nhấn nút play ở main menu 2. Người dùng chọn map được hiển thị 3. Chương trình chuyển sang scene mới cho phép người chơi map 4. Người chơi finish màn chơi 5. Client gửi kết quả lên cho server  * Dòng sự kiện khác:   4.1: người chơi thất bại không vượt màn được  5.1: client không gửi kết quả lên do người chơi thất bại |
| * Các yêu cầu đặc biệt: |
| * Thiết bị người chơi phải có kết nối internet |
| * Trạng thái hệ thống khi bắt đầu Use-case: |
| * Game đã được khởi động * Server đã được khởi động |
| * Trạng thái hệ thống khi kết thúc Use-case: |
| * Client hiển thị kết quả chơi game * Server update kết quả chơi game của map đó |

Bảng 3.6 Đặc tả use-case play game

* Rate map:

|  |
| --- |
| * Tóm tắt: |
| * Tác nhân: ngươi chơi game * Use-case này cho phép người sử dụng đánh giá 1 map nào đó |
| * Dòng sự kiện: |
| * Dòng sự kiện chính:  1. Người chơi chọn map muốn đánh giá 2. Người chơi đánh giá map 3. Client gửi request đánh giá map lên server 4. Server trả response tương ứng  * Dòng sự kiện khác:   4.1 response trả về rỗng do vấn đề kết nối mạng |
| * Các yêu cầu đặc biệt: |
| * Thiết bị người chơi phải có kết nối internet |
| * Trạng thái hệ thống khi bắt đầu Use-case: |
| * Game đã được khởi động * Server đã được khởi động |
| * Trạng thái hệ thống khi kết thúc Use-case: |
| * Client hiển thị kết quả rate map * Server update rating của map đó trong database |

Bảng 3.7 Đặc tả use-case rate map

### Các màn hình

#### Sơ đồ màn hình

Hình 3.2 Sơ đồ màn hình

#### Mô tả các màn hình

##### Màn hình chính

Giới thiệu

Là màn hình đầu tiên khi người dùng khởi động game



Hình 3.3 Màn hình chính (1)

|  |  |
| --- | --- |
| **STT** | **Ý nghĩa** |
| 1 | Nút Race: Vào màn hình bảng xếp hạng để chọn map chơi game |
| 2 | Nút Mapediter: Vào màn hình chọn tạo map, hoặc tiếp tục map đã tạo. |
| 3 | Nút Setting: xuất hiện thanh cài đặt âm thanh và thông tin về game |

Bảng 3.8 Bảng chú thích cho màn hình chính

Các layer

Hiện tại màn hình này chỉ có 1 layer duy nhất chứa background và các menu item

Cocos component sử dụng

Layer

Scene

Sprite

Menu item image

Director

Node

##### Màn hình bảng xếp hạng

Giới thiệu

Là màn hình hiển thị bảng xếp hạng hiện tại, được xếp theo thứ tự từ cao xuống thấp, hiện tại có 2 bảng là

* Bảng xếp hạng các map được rate cao nhất
* Bảng các map được upload gần nhất

****

Hình 3.4 Màn hình bảng xếp hạng

|  |  |
| --- | --- |
| **STT** | **Ý nghĩa** |
| 1 | Nút Back: Trở về màn hình chính. |
| 2 | TableViewCell: Bấm vào cell để vào chơi map mà mình muốn. |
| 3 | Nút NEW: Sang chế độ danh sách map mới nhất. |
| 4 | Nút TOP: Sang chế độ danh sách map có lượt rate và chơi nhiều nhất. |
| 5 | TableView: Lướt lên xuống để xem danh sách. |

Bảng 3.9 Bảng chú thích cho màn hình bảng xếp hạng

Các layer

Cocos component sử dụng

Layer

Scene

Sprite

Menu item image

Director

Node

HttpClient

UserDefault

##### Màn hình chọn thông tin map

Giới thiệu

Là màn hình cho phép người dùng lựa chọn tạo map mới hay edit map hiện tại

* Màn hình khi đã có sẵn map

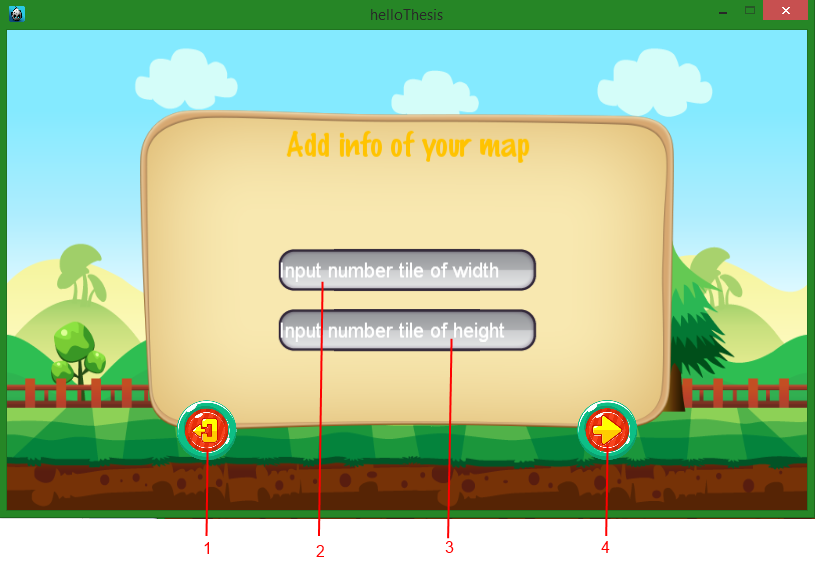
****

Hình 3.5 Màn hình khi có sẵn map

|  |  |
| --- | --- |
| **STT** | **Ý nghĩa** |
| 1 | Nút Back: Trở lại màn hình chính |
| 2 | Nút New: Tạo mới map, chuyển sang màn hình nhập thông số |
| 3 | Nút Next: Sang màn hình chỉnh sửa map với map đã lưu sẵn. |

Bảng 3.10 Bảng chú thích cho màn hình khi có sẵn map

* Màn hình thêm mới thông tin Map

****

Hình 3.6 Màn hình thêm mới map

|  |  |
| --- | --- |
| **STT** | **Ý nghĩa** |
| 1 | Nút Back: Trở về màn hình chính |
| 2 | EditBox Width: Nhập số tile chiều dài  + Chỉ được nhập số  + Số nhập vào lớn hơn 10 nhỏ hơn 500 |
| 3 | EditBox Height: Nhập số tile chiều cao  + Chỉ được nhập số  + Số nhập vào lớn hơn 10 nhỏ hơn 300 |
| 4 | Nút Tiếp tục: Chuyển sang màn hình chỉnh sửa map với thông số map vừa được tạo. |

Bảng 3.11 Bảng chú thích cho màn hình thêm mới map

Các layer

Màn hình này gồm 1 layer duy nhất

Cocos component sử dụng

Layer

Scene

Sprite

Menu item image

Director

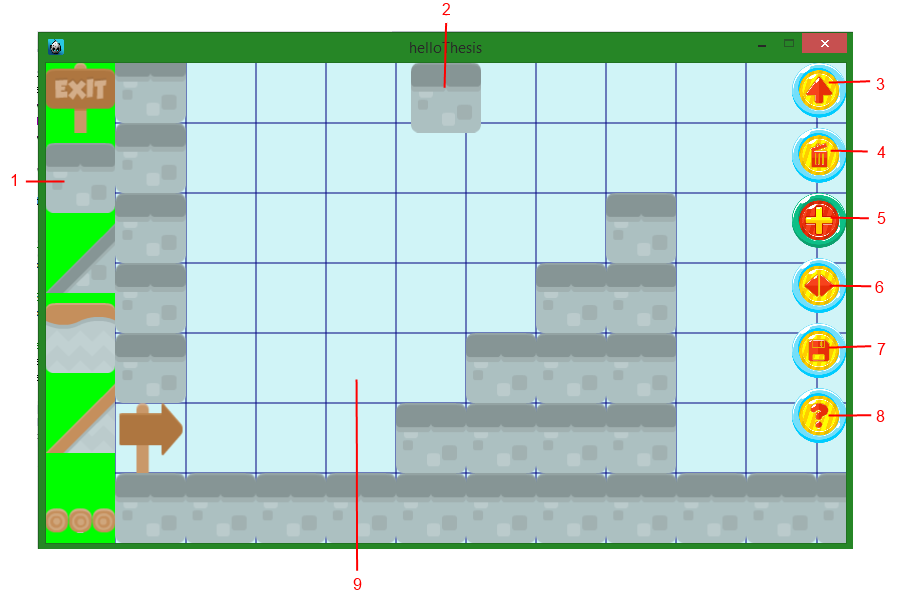
Node

UserDefault

##### Màn hình chỉnh sửa map

Giới thiệu

Là màn hình chính khi người chơi tương tác chỉnh sửa map



Hình 3.7 Màn hình chỉnh sửa map

|  |  |
| --- | --- |
| **STT** | **Ý nghĩa** |
| 1 | Hiển thị các đối tượng để thêm vào trong map. Bấm vào hình trong list view này để chọn đối tượng vẽ vào trong map. |
| 2 | Hình ảnh tượng trưng cho đối tượng đang được chọn |
| 3 | Nút Move: Chuyển về chế độ di chuyển. Trong chế độ này sẽ không thể xóa hoặc thêm đối tượng vào map, chỉ có thể vuốt để di chuyển map. |
| 4 | Nut Delete: Chuyển sang chế độ xóa. Trong chế độ này sẽ không thể di chuyển được map, hoặc thêm đối tượng vào map, chỉ có thể xóa đối tượng khi chạm vào đối tượng trên map. |
| 5 | Nút Add: Chuyển sang chế độ thêm mới. Trong chế độ này sẽ không thể di chuyển được map, hoặc xóa đối tượng trong map, chỉ có thể thêm đối tượng khi chạm vào ô bất kì trên map. |
| 6 | Nút Flip: lật đối tượng hình xiêng. |
| 7 | Nút Save: Lưu map ở bộ nhớ máy. Chuyển sang chế độ test map. |
| 8 | Nút Help: Chuyển sang hướng dẫn sử dụng map. |

Bảng 3.12 Bảng chú thích cho màn hình chỉnh sửa map

Các layer

Màn hình này chỉ có 1 layer duy nhất

Component cocos sử dụng

ScrollView: Toàn bộ map nằm trong này, lướt lên xuống để di chuyển tới vùng cần chỉnh sửa trong map.

DrawNode: Vẽ các đường thằng thông qua drawLine để tạo nên khung hình dạng lưới trong ScrollView

ListView: Chứa dánh sách các đối tượng thêm vào map, kéo lên xuống để sử dụng

Layer

Scene

Sprite

Menu item image

Director

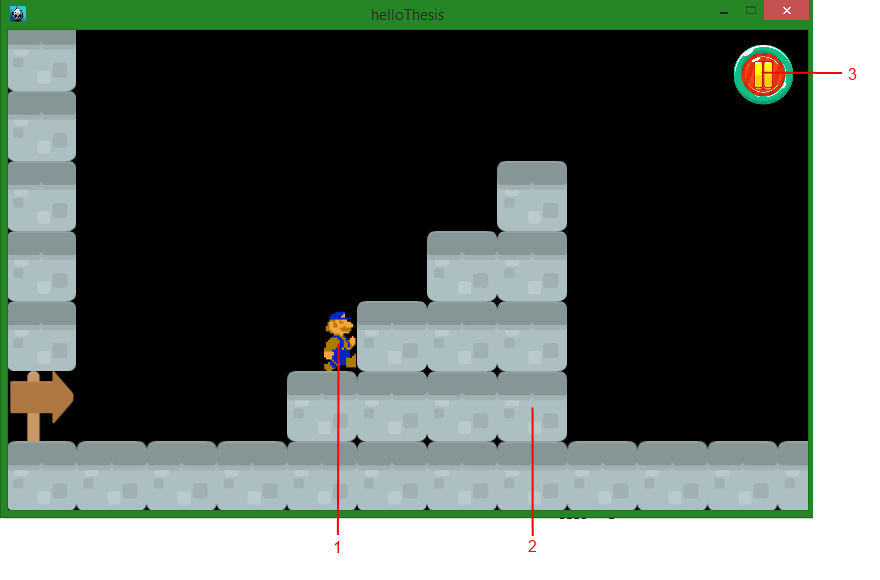
Node

##### Màn hình chơi game:

Giới thiệu

Là màn hình chính khi chơi game với thông tin map đã được load sẵn (từ server hay database local)

* Màn hình chơi game chính

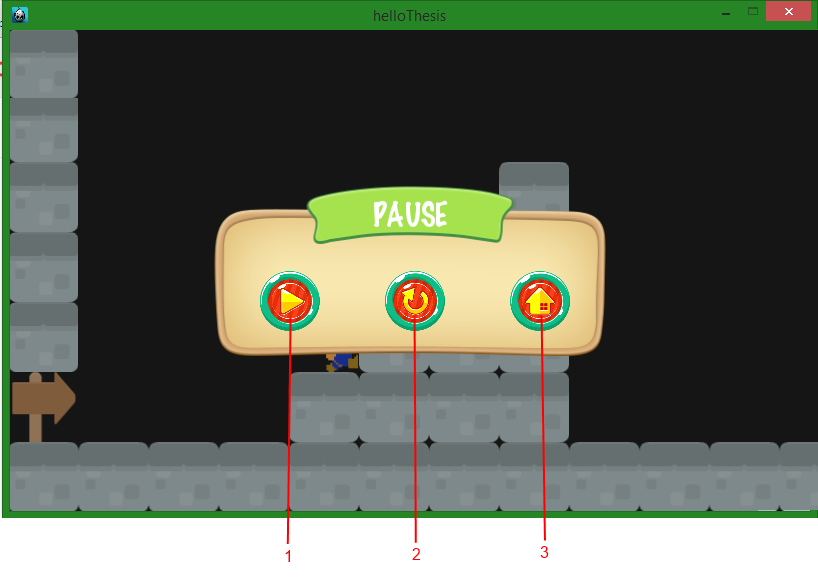
****

Hình 3.8 Màn hình chơi game chính

|  |  |
| --- | --- |
| **STT** | **Ý nghĩa** |
| 1 | Nhân vật di chuyển trong game. |
| 2 | Các đối tượng vật lý tương tác trong game với nhân vật. |
| 3 | Chuyển sang trạng thái tạm dùng chơi |

Bảng 3.13 Bảng chú thích cho màn hình chơi game chính

* Màn hình chơi game khi tạm dừng



Hình 3.9 Màn hình chơi game khi tạm dừng

|  |  |
| --- | --- |
| **STT** | **Ý nghĩa** |
| 1 | Nút Play: Tiếp tục chơi game |
| 2 | Nút Replay: Chơi lại game với map hiện tại. |
| 3 | Nút Home: Trở lại màn hình chính. |

* Bảng 3.16 Bảng chú thích cho màn hình chơi game khi tạm dừng
* Màn hình chơi game khi hoàn thành thử map



Hình 3.10 Màn hình chơi game khi hoàn thành thử map

|  |  |
| --- | --- |
| **STT** | **Ý nghĩa** |
| 1 | Nút Back: trở lại màn hình chọn thông tin map |
| 2 | Nút Upload: Tải map hiện tại lên server. |

Bảng 3.17 Bảng chú thích cho màn hình chơi game khi hoàn thành thử map

Các layer

Background: chứa background của game

Setting: chứa các button setting ingame

Object: chứa các object trong game: người chơi, vật cản, v.vv

PauseLayer: hiển thị các object khi người chơi pause game

Upload map: hiển thị b

Cocos component sử dụng

Box2d

Sprite

Scene

Layer

Event listener

Action

Camera

Node

Cocos studio integration

FileUtils

Menu item, menu

Director

Label

User default

http client

## Server-side

### Kiến trúc

#### So sánh các kiến trúc

Qua khảo sát, nhóm thu được bảng so sánh các kiến trúc sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Dễ sử dụng | Uyển chuyển | Dễ mở rộng | Dễ bảo trì | Bảo mật |
| Client-server | ✔ |  |  | ✔ | ✔ |
| Component based |  | ✔ | ✔ |  |  |
| N-tier | ✔ |  | ✔ | ✔ | ✔ |
| Message oriented |  |  | ✔ | ✔ | ✔ |
| Object oriented |  |  | ✔ |  |  |
| Service oriented |  |  | ✔ |  |  |

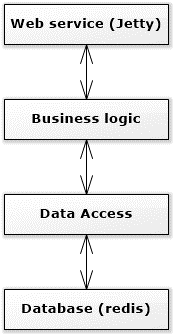
Bảng 3.2.1 So sánh các kiến trúc server

Qua bảng so sánh, nhóm quyết định chọn kiến trúc N-tier với những lý do sau:

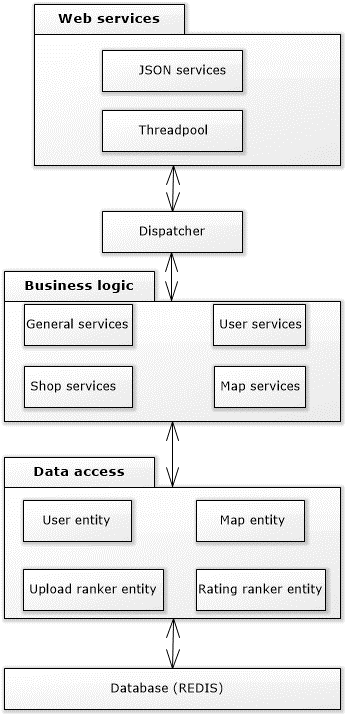
* Dễ sử dụng, mở rộng, bảo trì
* Quen thuộc
* Hiện tại chưa có kinh nghiệm với các kiến trúc còn lại

#### Mô hình kiến trúc server

##### Mô hình giản lược



##### Mô hình chi tiết



### Tầng web service

#### Cấu hình Jetty

* Version: 9.3
* Listening port: 9999
* Context path: /runner
* Thread pool: org.eclipse.jetty.util.thread.QueuedThreadPool
* Thread size:
  + Min: 100
  + Max: 2000
* Time out: 60 sec
* Handler: org.eclipse.jetty.servlet.ServletHandler

#### Cấu hình JSON response

* Header:
  + Access-Control-Allow-Origin 🡪 \*
  + Cache-Control 🡪 no-cache, no-store, must-revalidate
  + Pragma 🡪 no-cache
  + Expires 🡪 0
* Content type: application/json;charset=UTF-8
* Encoding : UTF-8

### Tầng business logic

Tầng này được thiết kế chia ra nhiều lớp như sau:

* User services: xử lý các request liên quan tới user, ví dụ: create user, login
* Map services: xử lý các request liên quan tới map, ví dụ như: upload map, rate map, comment map.
* Shop services: xử lý các request liên quan tới mua đồ trong game
* General services: xử lý các request không thuộc nhóm trên

### Tầng data access layer

#### Hash entity

Là các entity có số lượng thuộc tính cố định, bao gồm:

* User: lưu giữ các thông tin của user ví dụ như: user id, user name, tiền, avata.
* Map: lưu trữ các thông tin của 1 map trong game, ví dụ như: map id, creator id,

Các lệnh dùng để tương tác với database Redis:

* hgetall: trả về tất cả fields và values tương ứng của hash tại key truyền vào
* hmset: set fields và values tương ứng của hash tại key truyền vào

#### List entity

Lưu trữ danh sách có thứ tự, bao gồm

* Upload ranker entity: lưu trữ có thứ tự các map id theo điều kiện xếp hạng là thời gian upload, map càng mới thì thứ hạng càng cao và ngược lại.
* Rating ranker entity: lưu trữ có thứ tự các map id theo điều kiện xếp hạng là thời gian upload, map càng mới thì thứ hạng càng cao và ngược lại.

Các lệnh dùng để tương tác với database redis

rpush: chèn thêm 1 phần tử vào bên phải của list hiện tại

rpop: xóa 1 phần tử ở bên phải của list hiện tại

lrange: lấy n phần tử trong list ra

Lý giải nguyên nhân nhóm sử dụng list thay vì sorted set của redis: về cơ bản, list và sorted set của redis rất tương đồng với vector và set container của c++ như sau:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Add | Remove | Lookup |
| set | add - O(log(n)) | erase - O(log(n)) | set.find - O(log(n)) |
| sorted set | zadd - O(log(n)) | zrem - O(log(n)) | zrange - O(log(n)) |
| list | lpush/rpush - O(1) | lpop/rpop - O(1) | lrangeO(n) |
| vector | push\_back - O(1) | pop\_back - O(1) | std::find - O(n) |

Bảng 3.2.4 Bảng so sánh set, sorted set, list, vector

Theo đó, ta nên dùng set/ sorted set khi yêu cầu thỏa mãn những điều kiện sau:

* Số lượng phần tử có thể tăng lên rất lớn khiến cho có sự khác biệt đáng kể giữa O(n) và O(log(n)) 🡨 thỏa
* Số lần lookups tương đương với số lần add 🡨 không chắc chắn
* Các phần tử được add vào 1 cách random, chứ không phải là theo thứ tự 🡨 không chắc chắn
* Add và remove được thực hiện xen kẽ nhau 🡨 không thỏa do bảng xếp hạng cứ sau 1 khoảng thời gian mới update 1 lần (hiện đang là 2h)

Xét thấy thỏa ½ điều kiện, tuy nhiên, có nguyên nhân chính khác như sau, nếu xài sorted set của redis, mỗi lần muốn thêm/ xóa/ lấy thông tin records thì đều phải gọi tới database, khá tốn chi phí (với trường hợp database đặt ở máy khác, ta sẽ phải chờ thêm 1 khoảng thời gian do vấn đề đường truyền mạng), trong khi với list, ta có thể tạo ra 1 arraylist trên memory rồi lấy trực tiếp trên đó, nhanh hơn rất nhiều, nên nhóm quyết định dùng list thay cho sorted set

### Tầng database

#### Cấu hình redis

* Version: 2.8.8
* Port: 6379
* Time out: 5000ms
* Max active : 10
* Max wait: 10
* Max idle: 10

#### Danh sách các “bảng”

Mặc dù redis là 1 nosql database dạng key-value, tuy nhiên, để dễ quản lý, các key được tạo ra đều tuân theo 1 pattern, ví dụ như sau: giả sử có 1 bảng User với khóa chính là Uid, với cái field là Name, Avata, Password; thì với redis, ta sẽ tạo 1 key User là 1 list trong đó chứa các Uid, đồng thời cứ với mỗi Uid, ta tạo thêm 1 key mới có dạng User:(uid), là 1 hash, có các field là Name, Avata, Password.

Như vậy, với cấu trúc như trên, thì trong database có các “bảng” chính sau:

* gr:maps:(mapID): lưu trữ thông tin của map tương ứng với mapID
* gr:phone\_id\_mapping: lưu trữ thông tin mã máy tương ứng với user id nào
* gr:u:maps: lưu trữ danh sách các map mà user tạo ra
* gr:u:(uid): lưu trữ thông tin của user

# THỬ NGHIỆM

## QUY TRÌNH THỬ NGHIỆM.

### Các thành phần của hệ thống thử nghiệm.

Các thành phần cần chuẩn bị để thử nghiệm hệ thống bao gồm:

* Về phần mềm:
  + Visual studio 2013
  + Xcode
* Về phần cứng:
  + Một máy điện thoại sử dụng hệ điều hành android
  + Một máy điện thoại sử dụng hệ điều hành iOs

### Tiến hành thử nghiệm.

* Chạy game trên android
  + Tạo map mới
  + Upload map lên
  + Chơi map vừa tạo (với danh sách map từ server gửi về)
  + Share facebook
* Lặp lại tương tự với iOs

## MỘT VÀI HÌNH ẢNH **KẾT QUẢ THỬ NGHIỆM.**

### Màn hình chơi game

Map được test có kích thước là 500x500

#### Không sử dụng quadtree



Hình 4.1 Màn hình chơi game không sử dụng quadtree

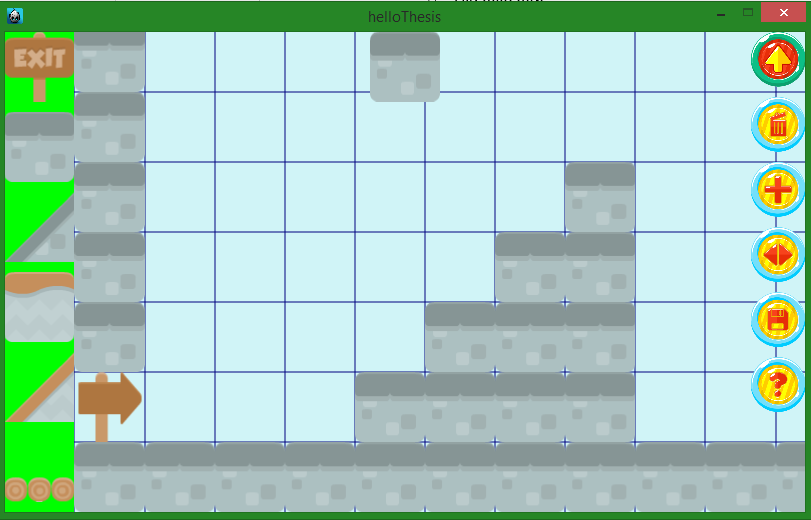
#### Sử dụng quadtree



Hình 4.2 Màn hình chơi game sử dụng quadtree

Dễ thấy, khi sử dụng quadtree, performance tốt hơn hẳn

### Màn hình chỉnh sửa map



Hình 4.3 Màn hình chỉnh sửa map

### Màn hình bảng xếp hạng map



Hình 4.4 Màn hình bảng xếp hạng map

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## KẾT QUẢ ĐẠT ĐƯỢC.

### Đối với đề tài

* Performance tốt, fps luôn ở mức 60
* Đã hoàn thành gameplay cơ bản
* Đã có thể tạo map và upload map lên
* Đã có thể download map về và chơi trên điện thoại của mình
* Đã có thể share facebook để thách thức bạn bè vào chơi

### Đối với các thành viên trong nhóm

* Hiểu được quy trình phát triển game.
* Hiểu được cấu trúc của engine Cocos2d-x.
* Biết cách xây dựng game từ engine Cocos2d-x.
* Ứng dụng được các kiến thức đã tìm hiểu được từ cocos2dx, game server vào việc thực hiện đồ án.
* Ôn luyện lại đồng thời bổ sung thêm kiến thức về ngôn ngữ C++ và Java
* Trau dồi thêm kỹ năng lập trình cũng như tư duy logic và khả năng xử lý vấn đề
* Kỹ năng làm việc nhóm của các thành viên được tăng cao, nhất là trong trường hợp 2 thành viên ở xa không thường xuyên gặp nhau được
* Khả năng quản lý, sắp xếp, phân chia công việc hợp lý được rèn luyện và hoàn thiện hơn

## Hạn chế

* Game play còn đơn giản
* Resource (hình ảnh, âm thanh) còn nhiều thiếu sót.
* Còn nhiều lỗi

## HƯỚNG PHÁT TRIỂN

### Về mặt chức năng

* Thêm chế độ multiplayer
* Tạo thêm hệ thống quest mới những map đã được tạo sẵn, để người chơi có thể chinh phục game từng phần, giúp người chơi gắn bó với game lâu hơn
* Thêm hệ thống shop cho phép người chơi dùng tiền (trong game hoặc ngoài) để mua thêm đồ đạc tạo map.

### Về mặt kỹ thuật

Config để có thể build ra bản cho window phone

## Những khó khăn và cách giải quyết

|  |  |
| --- | --- |
| Khó khăn | Giải quyết |
| Các thành viên chỉ là coder, không phải designer nên gặp khó khăn khi thiết kế game | Tham khảo các tool tương tự như mario maker + tham khảo ý kiến giảng viên hướng dẫn |
| Resource game không có | Cố gắng tìm các resource free trên mạng |
| Các thành viên không có điều kiện gặp nhau do ở xa + đi làm full time trên công ty | Làm việc, quản lý source code bằng github, sử dụng các công cụ để liên lạc khác như mạng xã hội, email để liên lạc bàn bạc, cố gắng gặp nhau cuối tuần |
| Chưa có kinh nghiệm làm game sever | Tìm kiếm tham khảo các project tương tự, tự customize lại cho phù hợp |

## Những bài học thu được

Qua đề tài, chúng em đã thu được những bài học quý báu, nổi bật nhất là:

* Không được chủ quan, luôn đánh giá đúng tình hình thực trạng
* Khi lên ý tưởng game, không lên quá mơ mộng xa vời dẫn tới trường hợp không thể thực hiện được trong thời gian yêu cầu 🡪 đầu voi đuôi chuột

.

TÀI LIỆU THAM KHẢO

[1] <http://www.cocos2d-x.org/wiki/Reference>

[2] <http://www.cocos2d-x.org/docs/README>

[3] <http://redis.io/commands>

[4] <http://www.eclipse.org/jetty/documentation/current/>

[5] <http://lafstern.org/matt/col1.pdf>

[6] <http://zeroturnaround.com/rebellabs/the-great-java-application-server-debate-with-tomcat-jboss-glassfish-jetty-and-liberty-profile/>

[7] <http://stackoverflow.com/questions/15108139/difference-between-socket-programming-and-http-programming>

[8] <http://www.slideshare.net/dvirsky/introduction-to-redis/>

[9] <http://msdn.microsoft.com/>

[10] <http://cocos2d-x.org/>

[11] <http://apache.org/licenses/LICENSE-2.0>

[12] <http://vi.wikipedia.org/>

[13] <http://en.wikipedia.org/>

[14] <http://www.fsf.org/licenses/gpl.html>

[15] Warcraft 3 world editor

[16] Mario maker Wii U