**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.

LỜI CẢM ƠN

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập ở giảng đường đại học đến nay, chúng em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý Thầy Cô, gia đình và bạn bè. Với lòng biết ơn sâu sắc nhất, chúng em xin gửi đến quý Thầy Cô ở Khoa Công Nghệ Phần Mềm – Trường Đại Học Công Nghệ Thông Tin đã cùng với tri thức và tâm huyết của mình để truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường.

Để hoàn thành luận văn này, chúng em xin chân thành gửi lời cảm ơn sâu sắc tới Thầy Phó giáo sư-Tiến sĩ Vũ Thanh Nguyên, người đã dành thời gian quý báu của mình để hướng dẫn trực tiếp chúng em cách thức nghiên cứu đề tài, để chúng em hoàn thành tốt đề tài của mình.

Trong quá trình làm luận văn, cũng như là trong quá trình làm bài báo cáo, khó tránh khỏi sai sót, rất mong các Thầy, Cô bỏ qua. Đồng thời do trình độ lý luận cũng như kinh nghiệm thực tiễn còn hạn chế nên bài báo cáo không thể tránh khỏi những thiếu sót, chúng em rất mong nhận được ý kiến đóng góp Thầy, Cô để chúng em học thêm được nhiều kinh nghiệm và sẽ hoàn thành tốt hơn bài báo cáo tốt nghiệp sắp tới.

CHÚNG EM XIN CHÂN THÀNH CẢM ƠN!

Sinh viên thực hiện

**Phạm Vũ Thanh Phong**

**Bành Lê Vũ Thiện**

Khoa Công Nghệ Phần Mềm, Lớp CNPM2011

NHẬN XÉT

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**………………………………………………………………………………………...**

**.......................................................................................................................................**

**.......................................................................................................................................**

MỤC LỤC

LỜI CẢM ƠN iv

NHẬN XÉT v

MỤC LỤC vi

DANH MỤC HÌNH x

DANH MỤC BẢNG xiv

DANH MỤC TỪ VIẾT TẮT xvii

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

1.3.1 Khách thể nghiên cứu: 4

1.3.2 Đối tượng nghiên cứu: 4

1.3.3 Phạm vi nghiên cứu: 5

1.4 Ý NGHĨA LÝ LUẬN VÀ THỰC TIỄN CỦA ĐỀ TÀI. 6

1.5 THUẬN LỢI VÀ KHÓ KHĂN. 6

1.5.1 Thuận lợi. 6

1.5.2 Khó khăn. 7

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ PHƯƠNG PHÁP NGHIÊN CỨU 8

2.1 CƠ SỞ LÝ LUẬN. 8

2.1.1 Giới thiệu sơ lược về GDI+. 8

2.1.1.1 Tổng quan. 8

2.1.1.2 Các đối tượng cơ bản. 9

2.1.1.2.1 Graphics 9

2.1.1.2.2 Color 11

2.1.1.2.3 Brush 11

2.1.1.2.4 Pen 12

2.1.1.2.5 Region 12

2.1.1.2.6 Image 13

2.1.1.3 Kỹ thuật di chuyển ảnh. 13

2.1.1.4 Kỹ thuật chống flicker (nháy ảnh) 14

2.1.1.4.1 Nguyên nhân 14

2.1.1.4.2 Cách xử lý 14

2.1.1.5 Hệ tọa độ 15

2.1.2 Giới thiệu sơ lược về Cocos2d-x. 16

2.1.2.1 Giới thiệu 16

2.1.2.2 Giấy phép 16

2.1.2.3 Các tính năng 17

2.1.2.4 Kiến trúc engine Cocos2d-x 19

2.1.2.4.1 CCScene 20

2.1.2.4.2 CCNode 22

2.1.2.4.3 CCAction 24

2.1.2.4.4 Scheduled 29

2.1.2.4.5 CCDirector 30

2.1.2.4.6 CCTransitionScene 31

2.2 PHƯƠNG PHÁP NGHIÊN CỨU. 32

2.2.1 Phương pháp nghiên cứu trực tiếp. 32

2.2.2 Phương pháp nghiên cứu gián tiếp. 33

2.3 CÁC KHÁI NIỆM LIÊN QUAN. 34

2.3.1 Tổng quan về Mã nguồn mở. 34

2.3.1.1. Giới thiệu mã nguồn mở. 34

2.3.1.2 Giới thiệu phần mềm mã nguồn mở 35

2.3.1.2.1 Khái niệm 35

2.3.1.2.2 Lợi ích của phần mềm mã nguồn mở 36

2.3.1.2.3 Hạn chế của phần mềm mã nguồn mở 40

2.3.2.1.4 Các loại phần mềm mã nguồn mở 42

2.3.1.3 Giới thiệu giấy phép mã nguồn mở 44

Giấy phép mã nguồn mở 44

Tính pháp lý của giấy phép mã nguồn mở 45

Phân loại giấy phép mã nguồn mở 46

2.3.1.3 Các loại giấy phép thông dụng 47

2.3.1.3.1 Giấy phép GNU 47

2.3.1.3.2 Giấy phép BSD 55

2.3.1.3.3 Giấy phép MIT 56

2.3.1.3.4 Giấy phép Apache 57

2.3.1.3.5 Giấy phép Artistic 59

2.3.1.4 Vần đề tương thích của các giấy phép 61

2.3.1.5 Chọn giấy phép phù hợp 64

CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG EASYDESIGN 65

3.1 PHÂN TÍCH XỬ LÝ 65

3.1.1 Xây dựng mô hình Use-case 65

3.1.2 Đặc tả Use-case 66

3.1.3 Sơ đồ tuần tự 69

3.1.4 Sơ đồ luồng dữ liệu (DFD) 71

3.2 THIẾT KẾ 76

3.2.1 Sơ đồ màn hình 76

3.2.2 Mô tả các màn hình 78

CHƯƠNG 4: THIẾT KẾ FRAMEWORK EDESIGN 94

4.1 KHÁI QUÁT VỀ FRAMEWORK: 94

4.2 CÁC MÔ HÌNH GIỮA FRAMEWORK EDESIGN VÀ COCOS2D-X 94

4.3 CÁC MÔ HÌNH GIỮA FRAMEWORK EDESIGN VÀ ỨNG DỤNG EASY DESIGN 99

4.4 THIẾT KẾ TÍNH NĂNG FRAMEWORK 102

4.4.1 Tính năng Hành vi 102

4.4.1.1 Lớp fwBehavior 103

4.4.1.2 Lớp fwSave 104

4.4.1.3 Lớp fwSolid 105

4.4.1.4 Lớp fwBackground 106

4.4.1.5 Lớp fwBound 107

4.4.1.6 Lớp fwDestroyOutside 108

4.4.1.7 Lớp fwFollow 109

4.4.1.8 Lớp fwWarp 110

4.4.1.9 Lớp fwDrap\_Drop 111

4.4.1.10 Lớp fwBulletMove 113

4.4.1.11 Lớp fw8Direction 113

4.4.1.12 Lớp fwRotate 115

4.4.1.13 Lớp fwPlatform 115

4.4.1.14 Lớp fwOrbit 117

4.4.2 Tính năng Hiệu ứng 118

4.4.2.1 Lớp fwEffect 118

4.4.2.2 Lớp fwFade 119

4.4.2.3 Lớp fwBlink 120

4.4.2.4 Lớp fwTint 121

4.4.2.5 Lớp fwGrid 122

4.4.2.6 Lớp fwProgress 123

4.4.3 Tính năng Diễn ảnh 124

4.4.4 Tính năng Xử lý đầu vào 125

4.4.4.1 Bàn phím 126

4.4.4.2 Chuột 127

4.4.4.3 Cảm ứng chạm 129

4.4.4.4 Cảm ứng gia tốc 131

4.4.5 Tính năng bản đồ 132

4.4.6 Tính năng Sự kiện xử lý 134

4.4.7 Tính năng Quản lý đối tượng 135

CHƯƠNG 5: KẾT QUẢ THỬ NGHIỆM VÀ ĐÁNH GIÁ 137

5.1 QUY TRÌNH THỬ NGHIỆM. 137

5.1.1 Các thành phần của hệ thống thử nghiệm. 137

5.1.2 Tiến hành thử nghiệm. 137

5.2 MỘT VÀI HÌNH ẢNH KẾT QUẢ THỬ NGHIỆM. 138

CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 143

6.1 KẾT QUẢ ĐẠT ĐƯỢC. 143

6.2 ƯU ĐIỂM VÀ NHƯỢC ĐIỂM 143

6.2.1 Ưu điểm: 143

6.2.2 Nhược điểm: 143

6.3 HƯỚNG PHÁT TRIỂN 143

TÀI LIỆU THAM KHẢO 144

Tiếng Việt: 144

Tiếng Anh: 144

Website tham khảo: 144

Phần mềm tham khảo: 145

DANH MỤC HÌNH

[Hình 2.1: Các đối tượng cơ bản của GDI+ 9](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224102)

[Hình 2.2 Sử dụng thuộc tính Graphics 10](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224103)

[Hình 2.3 Phương thức CreateGraphics() 10](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224104)

[Hình 2.4 Dẫn xuất từ Bitmap 10](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224105)

[Hình 2.5 Thuộc tính lớp Color 11](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224106)

[Hình 2.6 Phương thức lớp Color 11](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224107)

[Hình 2.7 Phân loại Brush 12](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224108)

[Hình 2.8 Ví dụ cho Pen 13](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224109)

[Hình 2.9 Kiểm soát bằng biến trạng thái 14](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224110)

[Hình 2.10 Ví dụ cho kỹ thuật di chuyển hình ảnh 15](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224111)

[Hình 2.11 Cách xử lý nháy ảnh 16](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224112)

[Hình 2.12 Hệ tọa độ trong GDI+ 17](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224113)

[Hình 2.13 Kiến trúc Cocos2d-x 21](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224114)

[Hình 2.14 Khám phá các lớp của Cocos2d-x 22](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224115)

[Hình 2.15 Chụp màn hình của một game Cocos2d-x 22](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224116)

[Hình 2.16 Sơ đồ cây của một màn 23](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224117)

[Hình 2.17 Vị trí của một nút con khi được thêm vào nút cha 24](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224118)

[Hình 2.18 Sơ đồ các lớp kế thừa từ CCNode 25](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224119)

[Hình 2.19 Sơ đồ lớp kế thừa từ CCAction 27](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224120)

[Hình 2.20 Sơ đồ lớp CCActionIntervel 28](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224121)

[Hình 2.21 Sơ đố các lớp kế thừa từ CCActionEase 29](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224122)

[Hình 2.22 Sơ đồ các lớp kế thừa từ CCTilesGrid3Daction 30](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224123)

[Hình 2.23 Sơ đồ các lớp kế thừa từ CCGrid3DAction 31](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224124)

[Hình 2.24 Hiệu ứng lật trang 31](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224125)

[Hình 2.25 Sơ đồ các lớp kế thừa từ CCActionInstant 32](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224126)

[Hình 2.26 Sơ đồ các lớp kế thừa từ CCtransitionScene 35](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224127)

[Hình 2.27 Logo giấy phếp GNU 52](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224128)

[Hình 2.28 Logo giấy phép Apache 65](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224129)

[Hình 3.1 Sơ đồ Use-case 73](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224130)

[Hình 3.2 Sơ đồ tuần tự 79](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224131)

[Hình 3.3 Sơ đồ DFD cấp 0 81](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224132)

[Hình 3.4 Sơ đồ DFD cấp 1 82](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224133)

[Hình 3.5 Sơ đồ DFD cấp 2 chức năng Add Object 83](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224134)

[Hình 3.6 Sơ đồ DFD cấp 2 chức năng Add Instance variable, Behavior, Effect 84](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224135)

[Hình 3.7 Sơ đồ DFD chức năng Add Events 85](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224136)

[Hình 3.8 Sơ đồ DFD cấp 2 chứ năng Export 86](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224137)

[Hình 3.9 Sơ đồ màn hình 87](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224138)

[Hình 3.11 Màn hình chính (2) 89](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224139)

[Hình 3.12 Màn hình thêm đối tượng 92](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224140)

[Hình 3.14 Màn hình thêm đối tượng Button 94](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224141)

[Hình 3.15 Màn hình thêm thuộc tính 95](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224142)

[Hình 3.16 Màn hình thêm hành vi 96](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224143)

[Hình 3.17 Mành hình thông tin hành vi 8 Deriction 97](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224144)

[Hình 3.18 Màn hình thông tin hành vi Bullet 98](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224145)

[Hình 3.19 Màn hình Data Manger 99](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224146)

[Hình 3.20 Màn hình Audio Manager 100](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224147)

[Hình 3.21 Màn hình thêm Event 101](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224148)

[Hình 3.22 Màn hình thông tin Event 102](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224149)

[Hình 3.23 Màn hình Thêm Action 103](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224150)

[Hình 4.1 Sơ đồ hành vi 114](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224151)

[Hình 4.2 Sơ đồ lớp fwBehavior 115](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224152)

[Hình 4.3 Sơ đồ lớp fwSave 116](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224153)

[Hình 4.4 Sơ đồ lớp fwSolid 117](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224154)

[Hình 4.5 Sơ đồ lớp fwBackground 118](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224155)

[Hình 4.6 Sơ đồ lớp fwBound 119](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224156)

[Hình 4.7 Sơ đồ lớp fwDestrouOutside 120](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224157)

[Hình 4.8 Sơ đồ lớp fwFollow 121](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224158)

[Hình 4.9 Sơ đồ lớp fwWarp 122](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224159)

[Hình 4.10 Sơ đồ lớp fwDrap\_Drop 123](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224160)

[Hình 4.11 Sơ đồ lớp fwBulletMove 124](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224161)

[Hình 4.12 Sơ đồ lớp fw8Direction 125](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224162)

[Hình 4.13 Sơ đồ lớp fwRotate 126](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224163)

[Hình 4.14 Sơ đồ lớp fwPlatform 127](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224164)

[Hình 4.15 Sơ đồ lớp fwOrbit 129](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224165)

[Hình 4.16 Sơ đồ hiệu ứng 130](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224166)

[Hình 4.17 Sơ đồ lớp fwEffect 130](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224167)

[Hình 4.18 Sơ đồ lớp fwFade 131](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224168)

[Hình 4.19 Sơ đồ lớp fwBlink 132](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224169)

[Hình 4.20 Sơ đồ lớp fwTint 134](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224170)

[Hình 4.21 Sơ đồ lớp fwGrid 135](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224171)

[Hình 4.22 Sơ đồ lớp fwProgress 136](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224172)

[Hình 4.23 Sơ đồ lớp fwAnimation 137](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224173)

[Hình 4.24 Sơ đồ tính năng xử lý đầu vào 138](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224174)

[Hình 4.25 Sơ đồ lớp fwKeybroad 139](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224175)

[Hình 4.26 Sơ đồ lớp fwMouse 140](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224176)

[Hình 4.27 Sơ đồ lớp fwTouch 142](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224177)

[Hình 4.28 Sơ đồ lớp fwAcceleration 144](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224178)

[Hình 4.29 Sơ đồ lớp fwMap 146](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224179)

[Hình 4.30 Sơ đồ lớp fwEvent 147](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224180)

[Hình 4.32 Sơ đồ lớp fwObjectManager 148](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224181)

[Hình 5.1 Màn hình tạo project mới 151](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224182)

[Hình 5.2 Màn hình thêm Sprite 152](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224183)

[Hình 5.3 Màn hình thêm Button 152](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224184)

[Hình 5.4 Màn hình xuất file mã nguồn 152](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224185)

[Hình 5.5 Màn hình file mã nguồn 153](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224186)

[Hình 5.6 Màn hình chạy bằng Cocos2d-x (1) 154](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224187)

[Hình 5.6 Màn hình chạy bằng Cocos2d-x (2) 155](file:///C:\Users\lynch\Downloads\Khóa%20luận.docx#_Toc408224188)

DANH MỤC BẢNG

Bảng 2.1 So sánh một số giấy phép mã nguồn mở 71

Bảng 3.1 Danh sách Actor 73

Bảng 3.2 Danh sách use-case 74

Bảng 3.3 Đặc tả use-case Add Object 75

Bảng 3.4 Đặc tả use-case Drag and drop 75

Bảng 3.5 Đặc tả use-case Add Behaviors 76

Bảng 3.6 Đặc tả use-case Add Effects 76

Bảng 3.7 Đặc tả use-case Add Instance variables 77

Bảng 3.8 Đặc tả use-case Add Events 78

Bảng 3.9 Đặc tả use-case Export 78

Bảng 3.10 Bảng chú thích cho màn hình chính 91

Bảng 3.11 Bảng chú thích cho màn hình thêm đối tượng 92

Bảng 3.12 Bảng chú thích cho màn hình thêm đối tượng Sprite 93

Bảng 3.13 Bảng chú thích cho màn hình thêm đối tượng Button 94

Bảng 3.14 Bảng chú thích cho màn hình thêm thuộc tính 95

Bảng 3.15 Bảng chú thích cho màn hình thêm hành vi 96

Bảng 3.16 Bảng chú thích cho màn hình thông tin hành vi 8 Deriction 98

Bảng 3.17 Bảng chú thích cho màn hình thông tin hành vi Bullet 99

Bảng 3.18 Bảng chú thích cho màn hình Data Manager 100

Bảng 3.19 Bảng chú thích cho màn hình Audio Manager 101

Bảng 3.20 Bảng chú thích cho màn hình thêm Event 102

Bảng 3.21 Bảng chú thích cho màn hình thông tin Event 103

Bảng 3.22 Bảng chú thích cho màn hình thêm Action 103

Bảng 3.23 Bảng chú thích cho màn hình thông tin Action 104

Bảng 4.1 Thuộc tính lớp fwBehavior 115

Bảng 4.2 Phương thức lớp fwBehavior 115

Bảng 4.3 Thuộc tính lớp fwSave 116

Bảng 4.4 Phương thức lớp fwSave 116

Bảng 4.5 Thuộc tính lớp fwSolid 117

Bảng 4.6 Phương thức lớp fwSolid 117

Bảng 4.7 Thuộc tính lớp fwBackground 118

Bảng 4.8 Phương thức lớp fwBackground 118

Bảng 4.9 Thuộc tính lớp fwBound 119

Bảng 4.10 Phương thức lớp fwBound 119

Bảng 4.11 Thuộc tính lớp fwDestroyOutside 120

Bảng 4.12 Phương thức lớp fwDestroyOutside 120

Bảng 4.13 Thuộc tính lớp fwFollow 121

Bảng 4.14 Phương thức lớp fwFollow 121

Bảng 4.15 Thuộc tính lớp fwWarp 122

Bảng 4.16 Phương thức lớp fwWarp 122

Bảng 4.17 Thuộc tính lớp fwDrap\_Drop 123

Bảng 4.18 Phương thức lớp fwDrap\_Drop 123

Bảng 4.19 Thuộc tính lớp fwBulletMove 124

Bảng 4.20 Phương thức lớp fwBulletMove 125

Bảng 4.21 Thuộc tính lớp fw8Direction 126

Bảng 4.22 Phương thức lớp fw8Direction 126

Bảng 4.23 Thuộc tính lớp fwRotate 127

Bảng 4.24 Phương thức lớp fwRotate 127

Bảng 4.25 Thuộc tính lớp fwPlatform 128

Bảng 4.26 Phương thức lớp fwPlatform 128

Bảng 4.27 Thuộc tính lớp fwOrbit 129

Bảng 4.28 Phương thức lớp fwOrbit 129

Bảng 4.29 Thuộc tính lớp fwEffect 131

Bảng 4.30 Phương thức lớp fwEffect 131

Bảng 4.31 Thuộc tính lớp fwFade 132

Bảng 4.32 Phương thức lớp fwFade 132

Bảng 4.34 Thuộc tính lớp fwBlink 133

Bảng 4.34 Phương thức lớp fwBlink 133

Bảng 4.35 Thuộc tính lớp fwTint 134

Bảng 4.36 Phương thức lớp Tint 134

Bảng 4.37 Thuộc tính lớp fwGrid 135

Bảng 4.38 Phương thức lớp fwGrid 135

Bảng 4.39 Thuộc tính lớp fwProgress 136

Bảng 4.40 Phương thức lớp fwProgress 137

Bảng 4.41 Thuộc tính lớp fwAnimation 137

Bảng 4.42 Phương thức lớp fwAnimation 138

Bảng 4.43 Thuộc tính lớp fwKeybroad 139

Bảng 4.44 Phương thức lớp fwKeybroad 140

Bảng 4.45 Thuộc tính lớp fwMouse 141

Bảng 4.46 Phương thức lớp fwMouse 141

Bảng 4.47 Thuộc tính lớp fwTouch 142

Bảng 4.48 Phương thức lớp fwTouch 144

Bảng 4.49 Thuộc tính lớp fwAcceleration 145

Bảng 4.50 Phương thức lớp fwAcceleration 145

Bảng 4.51 Thuộc tính lớp fwMap 146

Bảng 4.52 Phương thức lớp fwMap 146

Bảng 4.53 Thuộc tính lớp fwEvent 147

Bảng 4.54 Phương thức lớp fwEvent 148

Bảng 4.55 Thuộc tính lớp fwObjectManager 149

Bảng 4.56 Phương thức lớp fwObjectManager 149

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 |
| DFD | Data Flow Diagram | Sơ đồ luồng dữ liệu |

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

### 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

## Giới thiệu sơ lược về 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

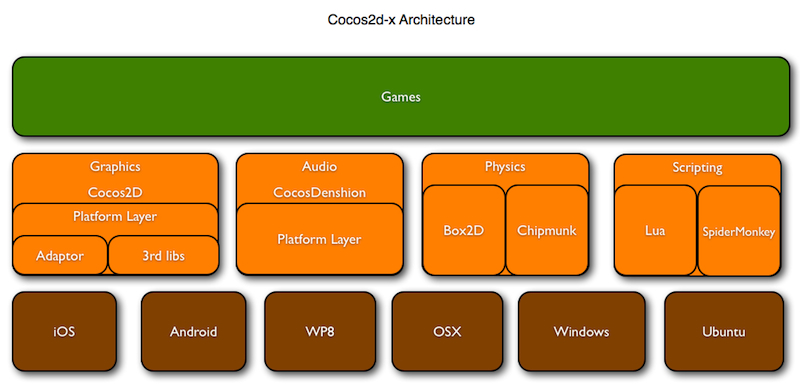
Cocos2d-x được sử dụng bởi các cá nhân và các tổ chức được biết như các công ty lớn. Ngày nay, nhiều game được viết bằng Cocos2d-x đang thống trị bảng doanh thu cao nhất của AppStore và Google Play, đặc biệt tại các quốc gia: Trung Quốc, Hàn Quốc và Nhật Bản. Những kỹ sư từ ChuKong, Google, Microsoft, 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

2.1.2.4 Kiến trúc engine Cocos2d-x

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



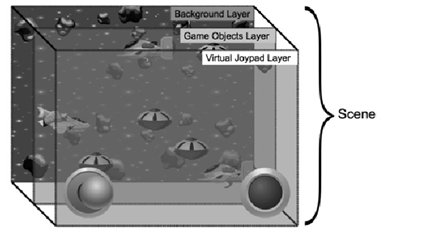
Hình 2.13 Kiến trúc Cocos2d-x

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.

2.1.2.4.1 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:



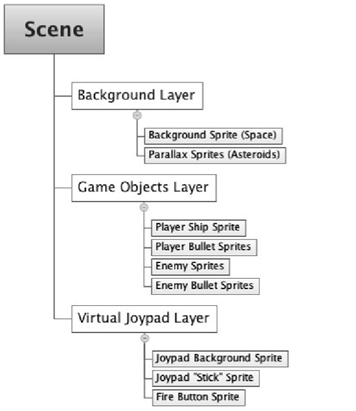
Hình 2.14 Khám phá các lớp của Cocos2d-x



Hình 2.15 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 2.16 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.

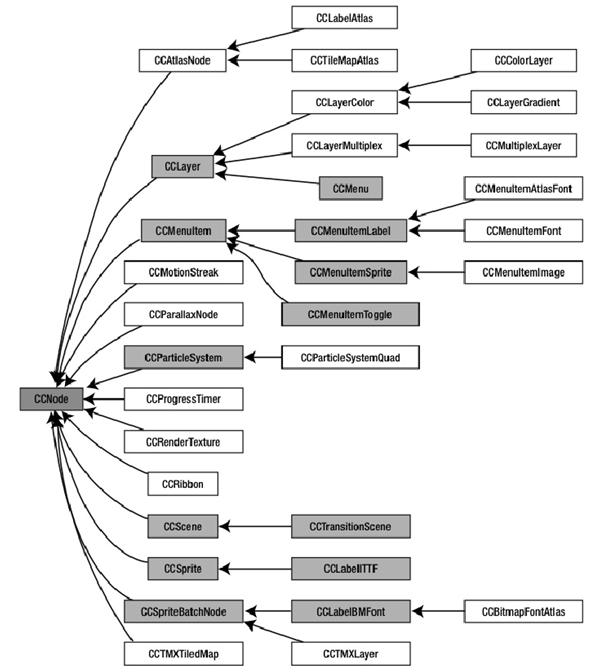
Tọa độ nút trong Cocos2d-X có liên quan đến tọa độ của nút cha, các nút con sẽ kế thừa các thuộc tính từ nút cha như tỉ lệ, độ xoay riêng màu sắc và độ hiển thị hình ảnh (Opacity, nằm trong khoảng 0 -> 255, tương ứng với alpha 0 -> 1).



Hình 2.17 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 2.18 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

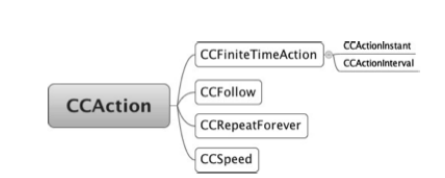
**Ngừng một hành động bằng định danh tag**:

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 2.19 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)

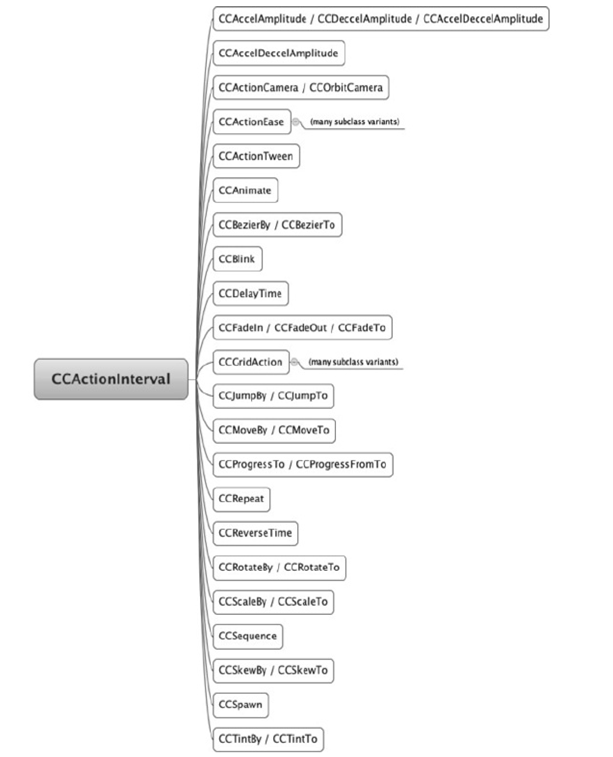
Label->runAction(CCFollow::create(player));

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.



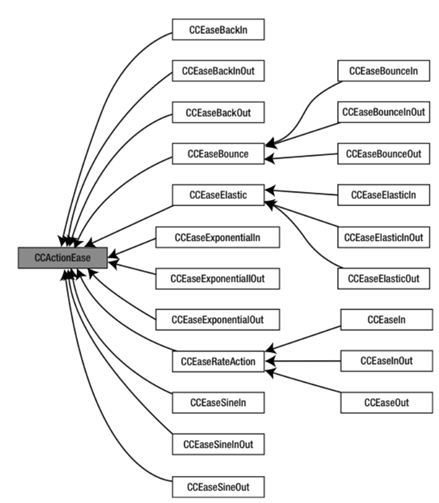
Hình 2.20 Sơ đồ lớp CCActionIntervel

**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));

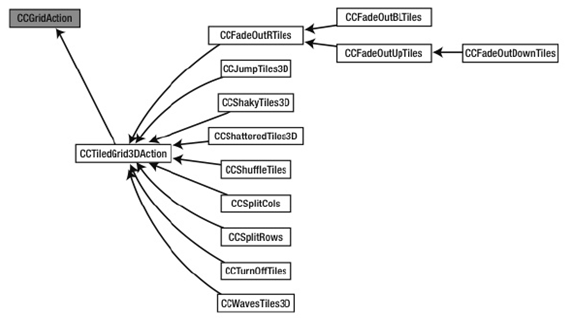
myCode->runAction(CCEaseInOut::create(move,4));



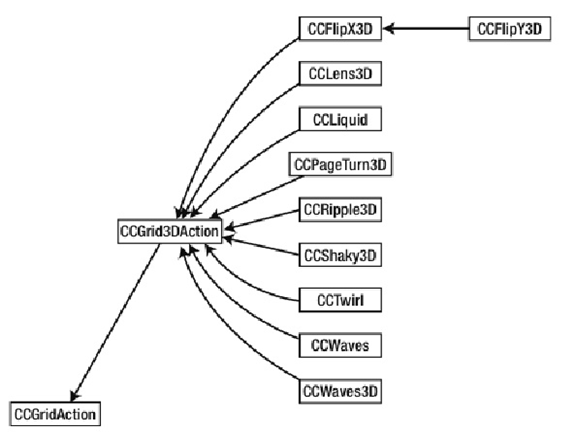
Hình 2.21 Sơ đố các lớp kế thừa từ CCActionEase

**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 2.22 Sơ đồ các lớp kế thừa từ CCTilesGrid3Daction



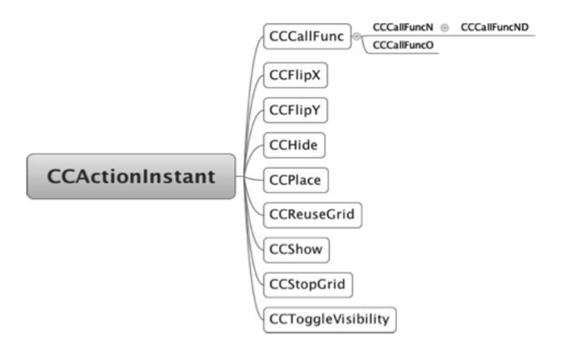
Hình 2.23 Sơ đồ các lớp kế thừa từ CCGrid3DAction

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 2.24 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 2.25 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.

2.3 CÁC KHÁI NIỆM LIÊN QUAN.

2.3.1 Tổng quan về Mã nguồn mở.

2.3.1.1. 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

2.3.1.2 Giới thiệu phần mềm mã nguồn mở

2.3.1.2.1 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.

2.3.1.2.2 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.

2.3.1.2.3 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.

2.3.2.1.4 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.

2.3.1.3 Giới thiệu giấy phép mã nguồn mở

Giấy phép mã nguồn mở

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.

2.3.1.3 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

2.3.1.3.1 Giấy phép GNU



Hình 2.27 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…

2.3.1.3.2 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.

2.3.1.3.3 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>

2.3.1.3.4 Giấy phép Apache

Description: ASF logo

Hình 2.28 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)

2.3.1.3.5 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.

2.3.1.4 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>

2.3.1.5 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:

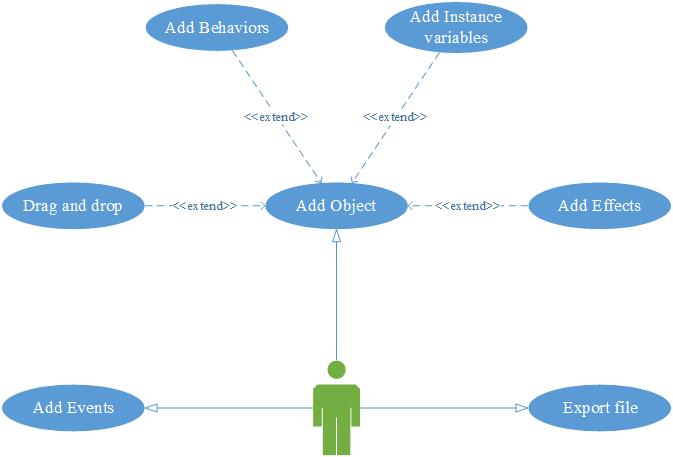
* Với những sản phẩm có dung lượng nhỏ, sử dụng đơn giản, nhà phát hành có thể sử dụng giấy phép BSD cho sản phẩm đó, vì BSD không bắt buộc phải công khai mã nguồn và hầu như không có người sử dụng nào muốn bỏ tiền ra để được bảo hành/đào tạo hướng dẫn sử dụng với một sản phẩm.
* Ngược lại, những sản 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

CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG EASYDESIGN

**3.1 PHÂN TÍCH XỬ LÝ**

3.1.1 Xây dựng mô hình Use-case



Hình 3.1 Sơ đồ Use-case

* Danh sách Actor:

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Actor** | **Ý nghĩa/ghi chú** |
| 1 | User | Người sử dụng chương trình Easy Design |

Bảng 3.1 Danh sách Actor

* Danh sách Use-case:

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Use-case** | **Ý nghĩa/ghi chú** |
| 1 | Add Object | Thêm đối tượng vào chương trình. |
| 2 | Drag and drop | Kéo thả đối tượng. |
| 3 | Add Behaviors | Thêm hành vi cho đối tượng. |
| 4 | Add Effects | Thêm hiệu ứng cho đối tượng. |
| 5 | Add Instance variables | Thêm thuộc tính cho đối tượng. |
| 6 | Add Events | Thêm sự kiện cho game. |
| 7 | Export file Cocos2d-x | Xuất file mã code theo cấu trúc Cocos2d-x |

Bảng 3.2 Danh sách use-case

3.1.2 Đặc tả Use-case

* Add Object:

|  |
| --- |
| * Tóm tắt: |
| * Tác nhân: ngươi sử dụng chương trình. * Use-case này cho phép người sử dụng chương trình thêm đối tượng vào chương trình. |
| * Dòng sự kiện: |
| * Dòng sự kiện chính:  1. Người dùng bấm chọn Add Object. 2. Chương trình hiển thị danh sách các loại Object. 3. Người dùng chọn loại Object muốn thêm vào. 4. Chương trình hiển thị thông tin Object cho người dùng điền vào. 5. Chương trình thêm Object vào danh sách Object và hiển thị lên màn hình.  * Dòng sự kiện khác:   Không có. |
| * Các yêu cầu đặc biệt: |
| * Không có. |
| * Trạng thái hệ thống khi bắt đầu Use-case: |
| * Project đã được khởi tạo. |
| * Trạng thái hệ thống khi kết thúc Use-case: |
| * Đối tượng được thêm vào danh sách đối tượng. * Đối tượng được vẽ lên màn hình. * Thanh Properties hiển thị thông tin đối tượng. |
| * Điểm mở rộng: |
| * Add Instance variables: thêm thuộc tính cho đối tượng. * Add Behaviors: thêm hành vi cho đối tượng. * Add Effects: thêm hiệu ứng cho đối tượng. |

Bảng 3.3 Đặc tả use-case Add Object

* Drag and drop:

|  |
| --- |
| * Tóm tắt: |
| * Tác nhân: ngươi sử dụng chương trình. * Use-case này cho phép người sử dụng chương trình kéo thả đối tượng từ vị trí này sang vị trí khác. |
| * Dòng sự kiện: |
| * Dòng sự kiện chính:  1. Người dùng chọn đối tượng. 2. Người dùng kéo đối tượng đến vị trí mới. 3. Chương trình cập nhật lại tọa độ cho đối tượng và hiển thị ở vị trí mới.  * Dòng sự kiện khác:   Không có. |
| * Các yêu cầu đặc biệt: |
| * Không có. |
| * Trạng thái hệ thống khi bắt đầu Use-case: |
| * Phải có đối tượng để chọn. |
| * Trạng thái hệ thống khi kết thúc Use-case: |
| * Đối tượng được vẽ lại tại vị trí mới. * Thông tin đối tượng được cập nhật lại. |
| * Điểm mở rộng: |
| * Không có. |

Bảng 3.4 Đặc tả use-case Drag and drop

* Add Behaviors:

|  |
| --- |
| * Tóm tắt: |
| * Tác nhân: ngươi sử dụng chương trình. * Use-case này cho phép người sử dụng chương trình thêm các hành vi cho đối tượng. |
| * Dòng sự kiện: |
| * Dòng sự kiện chính:  1. Người dùng chọn đối tượng. 2. Người dùng bấm nút Add Behaviors. 3. Chương trình hiển thị danh sách Behaviors cho người dùng chọn. 4. Người dùng chọn Behaviors muốn thêm cho đối tượng. 5. Chương trình hiển thị thông tin Behaviors cho người dùng chỉnh sửa. 6. Chương trình thêm Behaviors vào cho đối tượng.  * Dòng sự kiện khác:   Không có. |
| * Các yêu cầu đặc biệt: |
| * Không có. |
| * Trạng thái hệ thống khi bắt đầu Use-case: |
| * Phải có đối tượng để chọn. |
| * Trạng thái hệ thống khi kết thúc Use-case: |
| * Behaviors được thêm vào đối tượng và hiển thị. |
| * Điểm mở rộng: |
| * Delete Behaviors: Xóa Behaviors đã thêm vào đối tượng. |

Bảng 3.5 Đặc tả use-case Add Behaviors

* Add Effects:

|  |
| --- |
| * Tóm tắt: |
| * Tác nhân: ngươi sử dụng chương trình. * Use-case này cho phép người sử dụng chương trình thêm các hiệu ứng cho đối tượng. |
| * Dòng sự kiện: |
| * Dòng sự kiện chính:  1. Người dùng chọn đối tượng. 2. Người dùng bấm nút Add Effects. 3. Chương trình hiển thị thông tin Effects cho người dùng chỉnh sửa. 4. Chương trình thêm Effects vào cho đối tượng.  * Dòng sự kiện khác:   Không có. |
| * Các yêu cầu đặc biệt: |
| * Không có. |
| * Trạng thái hệ thống khi bắt đầu Use-case: |
| * Phải có đối tượng để chọn. |
| * Trạng thái hệ thống khi kết thúc Use-case: |
| * Effects được thêm vào đối tượng và hiển thị. |
| * Điểm mở rộng: |
| * Delete Effects: Xóa Effects đã thêm vào đối tượng. |

Bảng 3.6 Đặc tả use-case Add Effects

* Add Instance variables:

|  |
| --- |
| * Tóm tắt: |
| * Tác nhân: ngươi sử dụng chương trình. * Use-case này cho phép người sử dụng chương trình thêm các thuộc tính cho đối tượng. |
| * Dòng sự kiện: |
| * Dòng sự kiện chính:  1. Người dùng chọn đối tượng. 2. Người dùng bấm nút Add Instance variables. 3. Chương trình hiển thị thông tin Instance variables cho người dùng chỉnh sửa. 4. Chương trình thêm Instance variables vào cho đối tượng.  * Dòng sự kiện khác:   Không có. |
| * Các yêu cầu đặc biệt: |
| * Không có. |
| * Trạng thái hệ thống khi bắt đầu Use-case: |
| * Phải có đối tượng để chọn. |
| * Trạng thái hệ thống khi kết thúc Use-case: |
| * Instance variables được thêm vào đối tượng và hiển thị. |
| * Điểm mở rộng: |
| * Delete Instance variables: Xóa Instance variables đã thêm vào đối tượng. |

Bảng 3.7 Đặc tả use-case Add Instance variables

* Add Events:

|  |
| --- |
| * Tóm tắt: |
| * Tác nhân: ngươi sử dụng chương trình. * Use-case này cho phép người sử dụng chương trình thêm các sự kiện cho game. |
| * Dòng sự kiện: |
| * Dòng sự kiện chính:  1. Người dùng chọn tab Event. 2. Người dùng bấm nút Add Event. 3. Chương trình hiển thị danh sách đối tượng hỗ trợ Event. 4. Người dùng chọn đối tượng muốn thêm Event. 5. Chương trình hiển thị danh sách Event đối tượng hỗ trợ. 6. Người dùng chọn Event muốn thêm vào. 7. Chương trình thêm Event và hiển thị. 8. Người dùng bấm nút Add Action để thêm Action vào Event. 9. Chương trình hiển thị danh sách đối tượng hỗ trợ Action. 10. Người dùng chọn đố itượngmuốn thêm Action. 11. Chương trình hiển thị danh sách Action đối tượng hỗ trợ. 12. Người dùng chọn Action muốn thêm vào Event. 13. Chương trình thêm Action và hiển thị.  * Dòng sự kiện khác:   Không có. |
| * Các yêu cầu đặc biệt: |
| * Không có. |
| * Trạng thái hệ thống khi bắt đầu Use-case: |
| * Project đã được khởi tạo. |
| * Trạng thái hệ thống khi kết thúc Use-case: |
| * Danh sách Event được thêm vào và hiển thị. |
| * Điểm mở rộng: |
| * Delete Event: Xóa Event trong danh sách Event. |

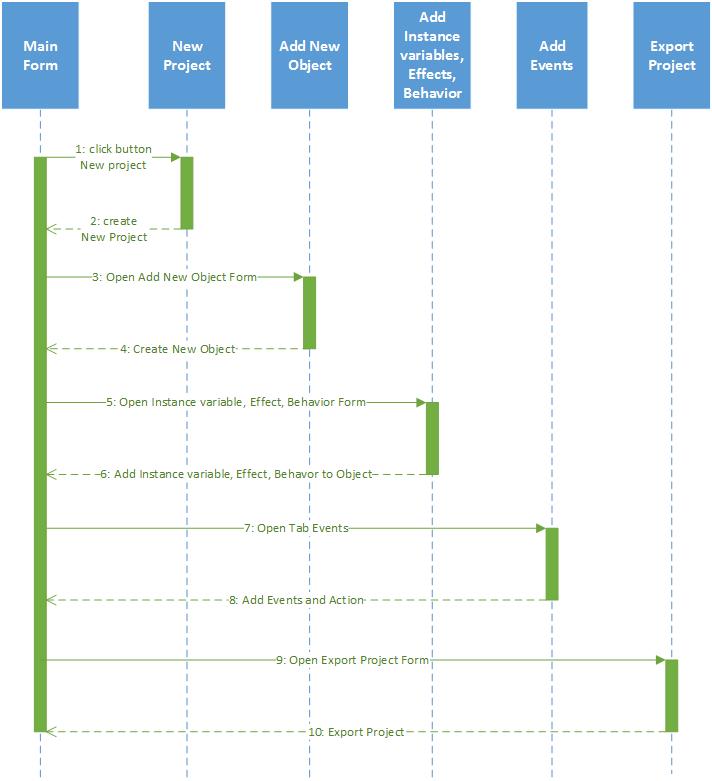
Bảng 3.8 Đặc tả use-case Add Events

* Export file Cocos2d-x:

|  |
| --- |
| * Tóm tắt: |
| * Tác nhân: ngươi sử dụng chương trình. * Use-case này cho phép người sử dụng chương trình xuất ra file code theo cấu trúc framework Cocos2d-x dựng sẵn. |
| * Dòng sự kiện: |
| * Dòng sự kiện chính:  1. Người dùng bâm nút Export. 2. Chương trình hiển thị form để người dùng lựa chọn vị trí xuất file mã nguồn. 3. Chương trình xuất file theo cấu trúc framework.  * Dòng sự kiện khác:   Không có. |
| * Các yêu cầu đặc biệt: |
| * Không có. |
| * Trạng thái hệ thống khi bắt đầu Use-case: |
| * Project đã được khởi tạo. |
| * Trạng thái hệ thống khi kết thúc Use-case: |
| * Xuất ra các file mã code. |
| * Điểm mở rộng: |
| * Không có. |

Bảng 3.9 Đặc tả use-case Export

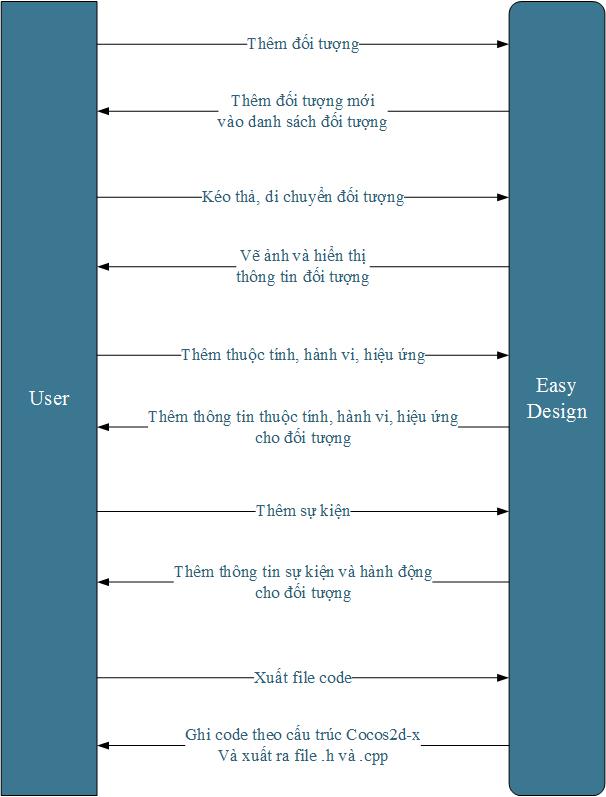
3.1.3 Sơ đồ tuần tự



Hình 3.2 Sơ đồ tuần tự

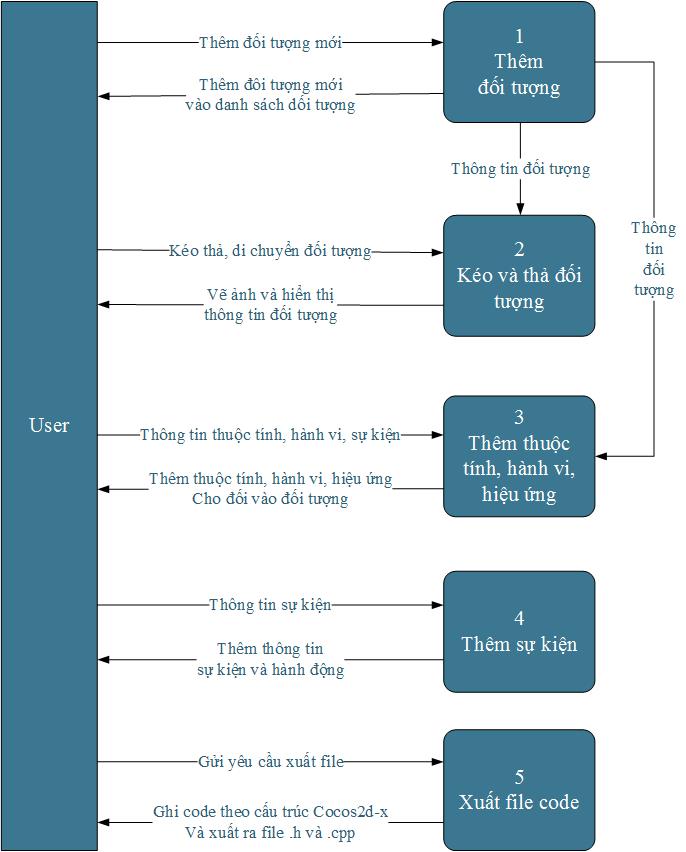
3.1.4 Sơ đồ luồng dữ liệu (DFD)

* Sơ đồ DFD cấp 0

****

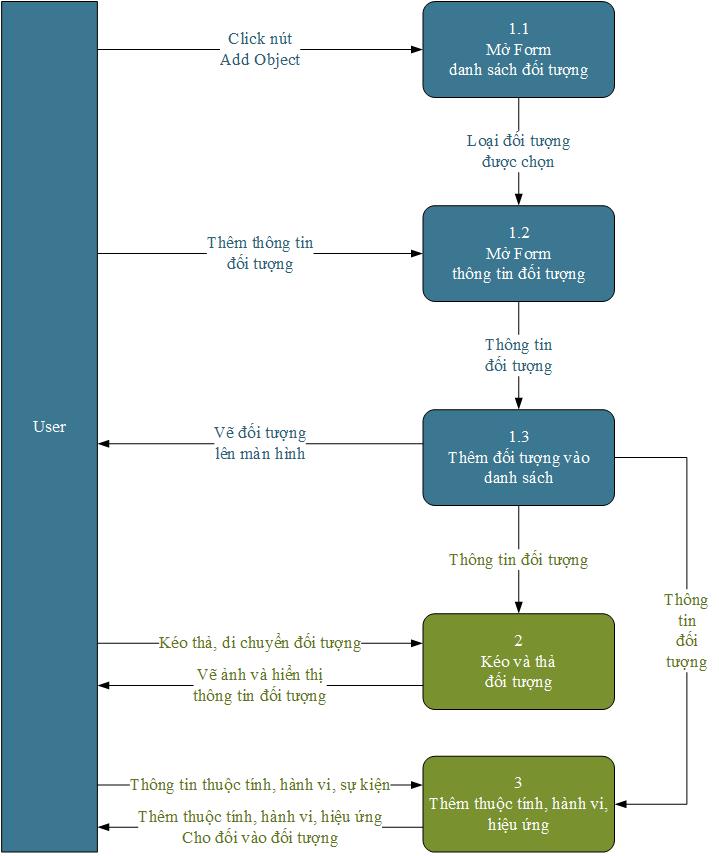
Hình 3.3 Sơ đồ DFD cấp 0

* Sơ đồ DFD cấp 1

****

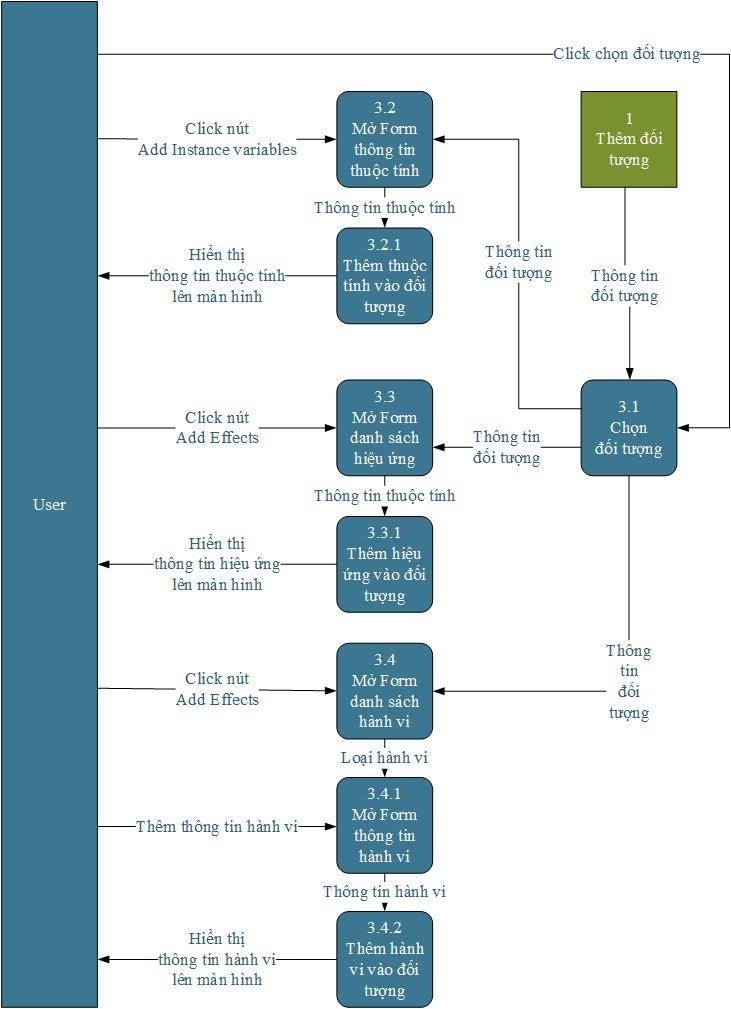
Hình 3.4 Sơ đồ DFD cấp 1

* Sơ đồ DFD cấp 2 chức năng Add Object

****

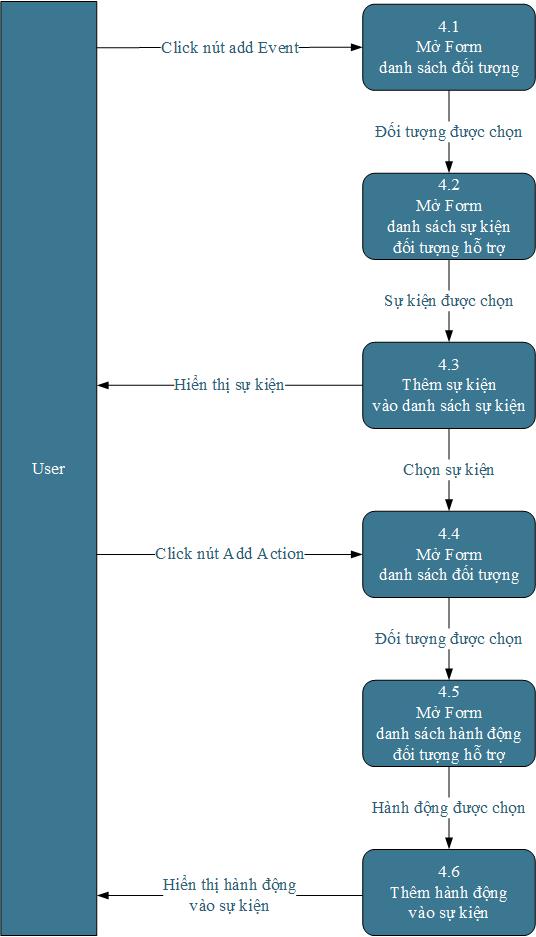
Hình 3.5 Sơ đồ DFD cấp 2 chức năng Add Object

* Sơ đồ DFD cấp 2 chức năng Add Instance variables, Behaviors, Effects

****

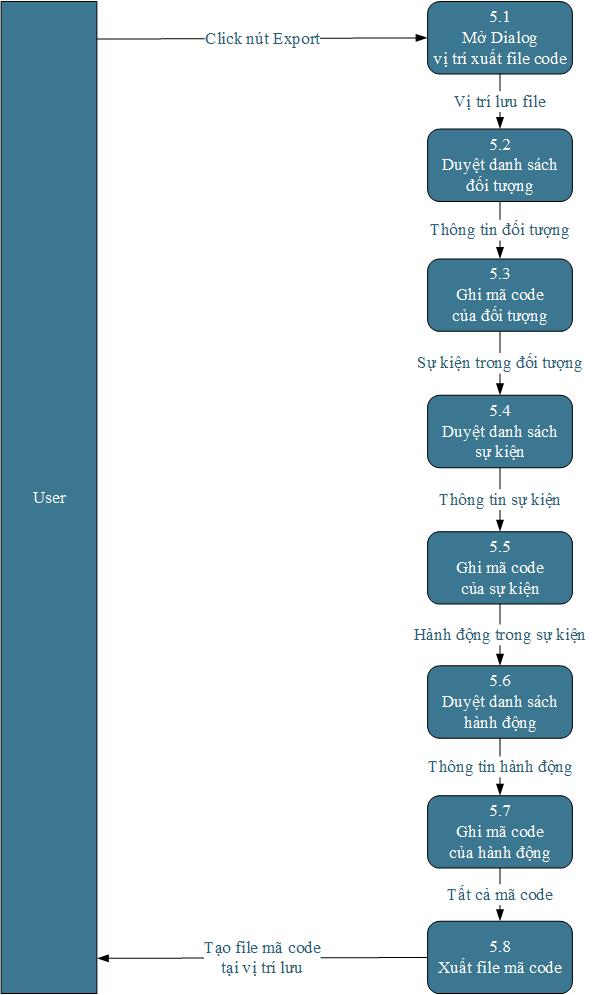
Hình 3.6 Sơ đồ DFD cấp 2 chức năng Add Instance variable, Behavior, Effect

* Sơ đồ DFD cấp 2 chức năng Add Events

****

Hình 3.7 Sơ đồ DFD chức năng Add Events

* Sơ đồ DFD cấp 2 chức năng Export

****

Hình 3.8 Sơ đồ DFD cấp 2 chứ năng Export

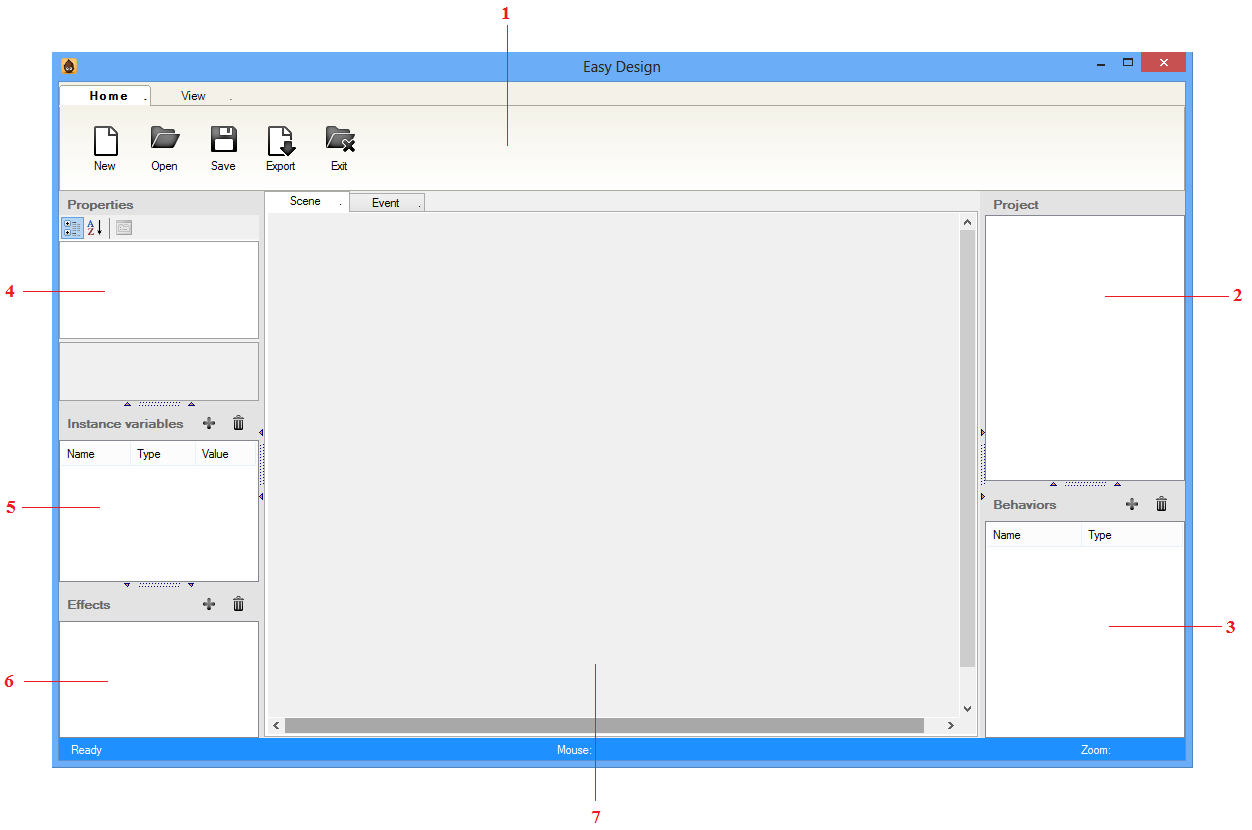
**3.2 THIẾT KẾ**

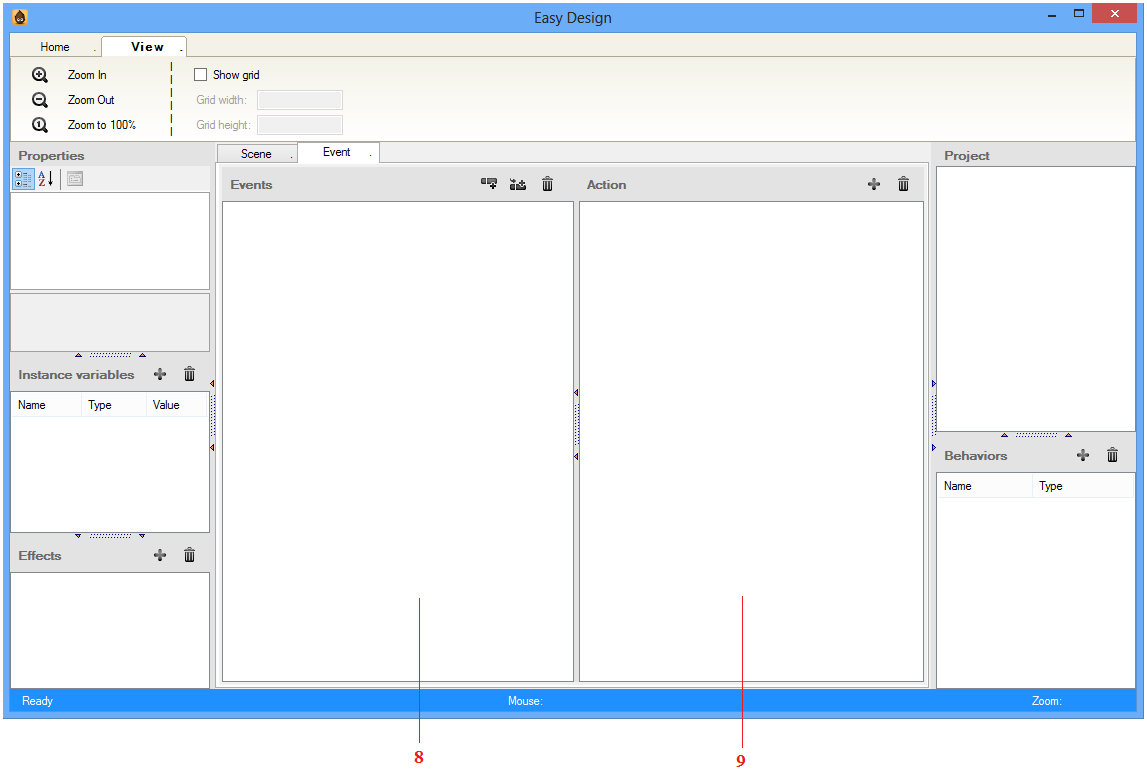
3.2.1 Sơ đồ màn hình

Hình 3.9 Sơ đồ màn hình

3.2.2 Mô tả các màn hình

* Màn hình chính

Hình 3.10 Màn hình chính (1)

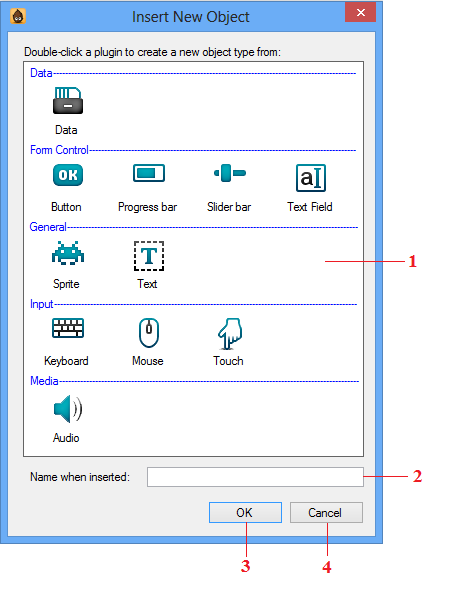


Hình 3.11 Màn hình chính (2)

|  |  |
| --- | --- |
| **STT** | **Ý nghĩa** |
| 1 | Thanh công cụ gồm các nút:   * New: Tạo một Project mới. * Open: Mở một Project có sẵn. * Save: Lưu Project đang làm. * Export: Xuất ra file mã nguồn. * Exit: Thoát chương trình. * ZoomIn: Phóng to ảnh lên gấp 2 lần. * ZoomOut: Thu nhỏ ảnh lại 1/2. * Zoom100: Trả ảnh lại về với kích thước ban đầu. * Grid: Hiển thị các được thẳng ngang dọc giúp việc căn chỉnh dễ dàng. |
| 2 | Hiển thị các thành phần của Project đang thực hiện theo dạng cây giúp thao tác qua lại giữa các thành phần trong Project |
| 3 | Hiển thị danh sách các hành vi đã thêm vào cho đối tượng. Có 2 nút chức năng:   * Add Behavior: Mở Form danh sách hành vi chương trình hỗ trợ. * Delete Behavior: Xóa 1 hành vi đang được chọn trong danh sách hiển thị. |
| 4 | Hiển thị các thuộc tính của đối tượng được chọn. Hỗ trợ việc thay đổi thuộc tính của đối tượng một cách nhanh chóng. |
| 5 | Hiển thị danh sách các thuộc đã thêm vào cho đối tượng. Có 2 nút chức năng:   * Add Instance variable: Mở Form thông tin thuộc để thêm thông tin thuộc tính vào đối tượng. * Delete Instance variable: Xóa 1 thuộc tính đang được chọn trong danh sách hiển thị. |
| 6 | Hiển thị danh sách các hiệu ứng đã thêm vào cho đối tượng. Có 2 nút chức năng:   * Add Effect: Mở Form danh sách hiệu ứng chương trình hỗ trợ. * Delete Effect: Xóa một hiệu ứng đang được chọn trong danh sách hiển thị. |
| 7 | Vùng để vẽ các đối tượng được thêm vào lên màn hình |
| 8 | Hiển thị danh sách các sự kiện đã thêm vào trong game. Có 3 nút chức năng:   * Add Event: Mở Form danh sách đối tượng để thêm sự kiện cho đối tượng. * Add Sub Event: tương tự như nút “Add Event” nhưng sự kiện này được thêm vào làm sự kiện con cho sự kiện được chọn. * Delete Event: Xóa một sự kiện đang được chọn trong danh sách hiển thị. |
| 9 | Hiển thị danh sách các hành động đã thêm vào trong game. Có 2 nút chức năng:   * Add Action: Mở Form danh sách đối tượng để thêm hành động cho đối tượng. * Delete Action: Xóa một hành động đang được chọn trong danh sách hiển thị. |

Bảng 3.10 Bảng chú thích cho màn hình chính

* Màn hình thêm đối tượng

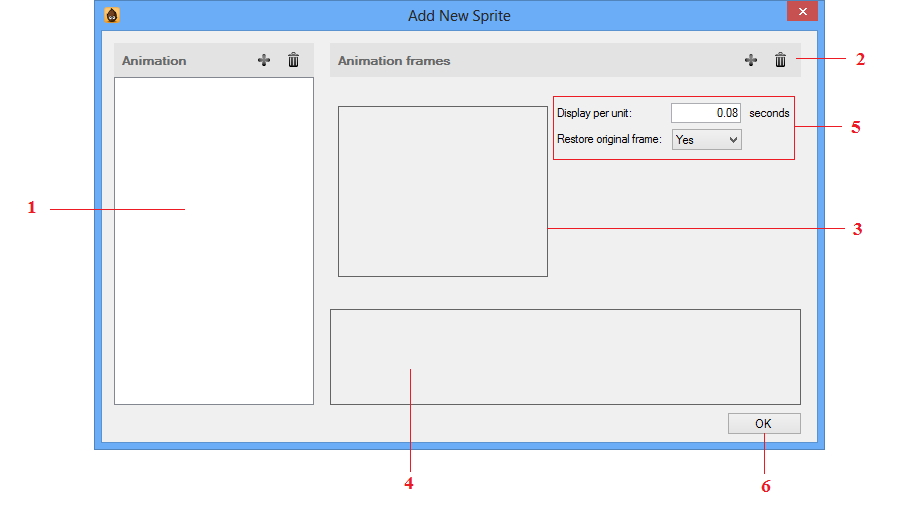


Hình 3.12 Màn hình thêm đối tượng

|  |  |
| --- | --- |
| **STT** | **Ý nghĩa** |
| 1 | Danh sách các đối tượng chương trình hỗ trợ |
| 2 | Đặt tên cho đối tượng. |
| 3 | Đồng ý thêm đối tượng vào chương trình. |
| 4 | Không đồng ý thêm đối tượng vào chương trình. |

Bảng 3.11 Bảng chú thích cho màn hình thêm đối tượng

* Màn hình thông tin đối tượng
* Màn hình thêm thông tin Sprite

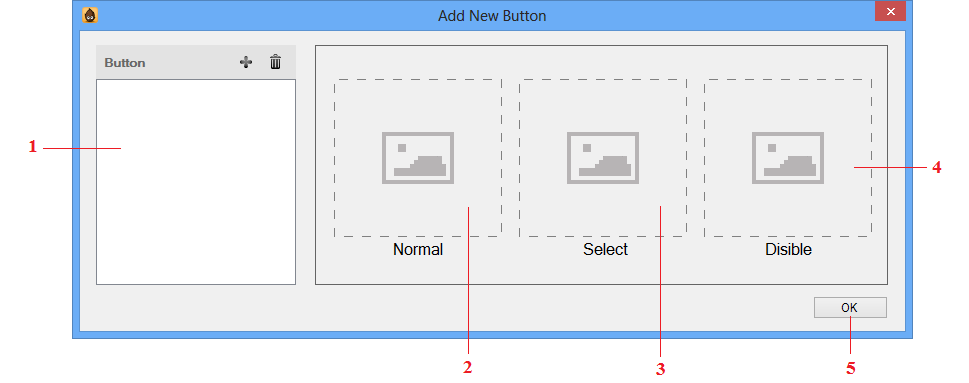


Hình 3.13 Màn hình thêm đối tượng Sprite

|  |  |
| --- | --- |
| **STT** | **Ý nghĩa** |
| 1 | Hiển thị danh sách các Animation đã thêm vào cho đối tượng. Có 2 nút chức năng:   * Add Animation: Thêm một Animation trống vào danh sách các Animation của đối tượng. * Delete Animation: Xóa một Animation đang được chọn trong danh sách các Animation của đối tượng. |
| 2 | Có 2 nút chức năng:   * Add Frame: Thêm một Frame mới vào trong Animation. * Delete Frame: Xóa một Frmae đang được chọn trong Animation. |
| 3 | Hiển thị hình ảnh của một frame được chọn. |
| 4 | Hiển thị danh sách các frame trong Animation. |
| 5 | Tùy chỉnh thông số cho Animation. Có 2 thông số:   * Display per unit: thời gian chuyển giữa 2 frame, đơn vị tính là giây. * Restore original frame: quay trở về frame ban đầu. |
| 6 | Lưu thông tin đối tượng và tắt form, trở về với form chính. |

Bảng 3.12 Bảng chú thích cho màn hình thêm đối tượng Sprite

* Màn hình thêm thông tin Button

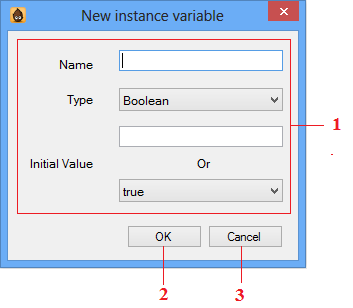


Hình 3.14 Màn hình thêm đối tượng Button

|  |  |
| --- | --- |
| **STT** | **Ý nghĩa** |
| 1 | Hiển thị danh sách các Button đã thêm vào cho đối tượng. Có 2 nút chức năng:   * Add Button: Thêm một Button trống vào danh sách các Button của đối tượng. * Delete Button: Xóa một Button đang được chọn trong danh sách các Button của đối tượng. |
| 2 | Trạng thái hiển thị bình thường của Button. |
| 3 | Trạng thái hiển thị khi Button được nhấn. |
| 4 | Trạng thái hiển thị khi Button bị vô hiệu hóa. |
| 5 | Lưu thông tin đối tượng và tắt form, trở về với form chính. |

Bảng 3.13 Bảng chú thích cho màn hình thêm đối tượng Button

* Màn hình thêm Instance variables

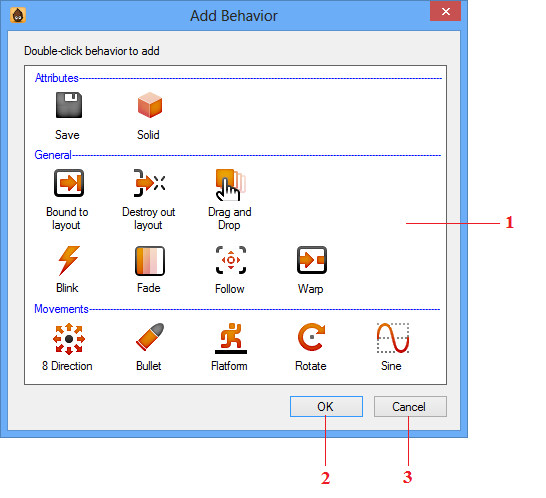


Hình 3.15 Màn hình thêm thuộc tính

|  |  |
| --- | --- |
| **STT** | **Ý nghĩa** |
| 1 | Hiển thị thông tin của thuộc tính sẽ thêm vào cho đối tượng. Các thông số bao gồm:   * Name: Tên thuộc tính sẽ thêm vào đối tượng. * Type: Kiểu thuộc tính sẽ thêm vào đối tượng. Chương trình hỗ trợ 3 kiểu thuộc tính: kiểu số, kiểu chữ, kiểu đúng sai. * Initial Value: Giá trị của thuộc tính. |
| 2 | Đồng ý thêm thuộc tính vào đối tượng. |
| 3 | Không đồng ý thêm thuộc tính vào đối tượng. |

Bảng 3.14 Bảng chú thích cho màn hình thêm thuộc tính

* Màn hình thêm Behaviors

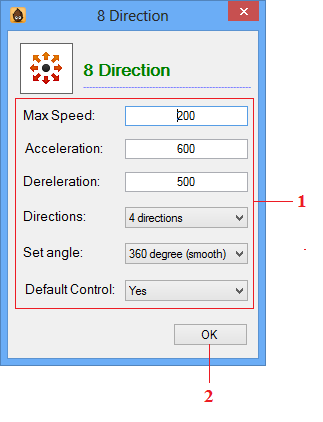


Hình 3.16 Màn hình thêm hành vi

|  |  |
| --- | --- |
| **STT** | **Ý nghĩa** |
| 1 | Danh sách các hành vi chương trình hỗ trợ cho đối tượng. |
| 2 | Đồng ý thêm hành vi vào đối tượng. |
| 3 | Không đồng ý thêm vi vào đối tượng. |

Bảng 3.15 Bảng chú thích cho màn hình thêm hành vi

* Màn hình thêm thông tin Behaviors
* Màn hình thông tin 8 Direction

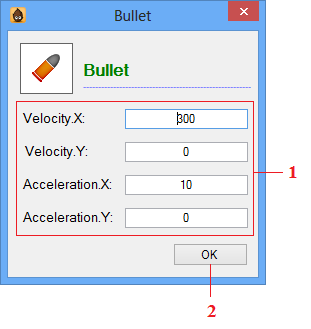


Hình 3.17 Mành hình thông tin hành vi 8 Deriction

|  |  |
| --- | --- |
| **STT** | **Ý nghĩa** |
| 1 | Hiển thị thông tin của hành vi sẽ thêm vào cho đối tượng. Các thông số bao gồm:   * Max Speed: Vận tốc tối đa đối tượng đạt được. * Acceleration: Gia tốc của đối tượng. * Dereleration: Gia tốc hãm của đối tượng. * Directions: Cách thay đổi góc di chuyển của đối tượng. Có 4 cách cơ bản: 45 độ, 90 độ, 360 độ và không thay đổi góc. * Angle: Cách xử lý di chuyển cho đối tượng. Có 4 cách cơ bản: lên xuống, trái phải, 4 hướng, 8 hướng. * Control: Đối tượng có điều khiển ngay đầu game. |
| 2 | Lưu thông tin của hành vi vào đối tượng và tắt form, trở về form chính. |

Bảng 3.16 Bảng chú thích cho màn hình thông tin hành vi 8 Deriction

* Màn hình thông tin Bullet

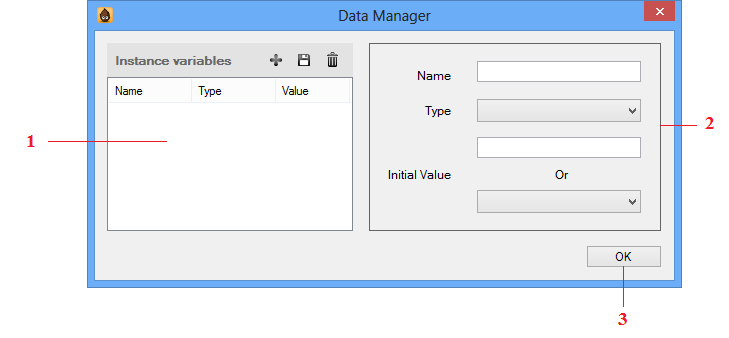


Hình 3.18 Màn hình thông tin hành vi Bullet

|  |  |
| --- | --- |
| **STT** | **Ý nghĩa** |
| 1 | Hiển thị thông tin của hành vi sẽ thêm vào cho đối tượng. Các thông số bao gồm:   * VelocityX: Vận tốc của đối tượng theo trục X. * VelocityY: Vận tốc của đối tượng theo trục Y. * AccelerationX: Gia tốc của đối tượng theo trục X. * AccelerationY: Gia tốc của đối tượng theo trục Y. |
| 2 | Lưu thông tin của hành vi vào đối tượng và tắt form, trở về form chính. |

Bảng 3.17 Bảng chú thích cho màn hình thông tin hành vi Bullet

* Màn hình Data Manager

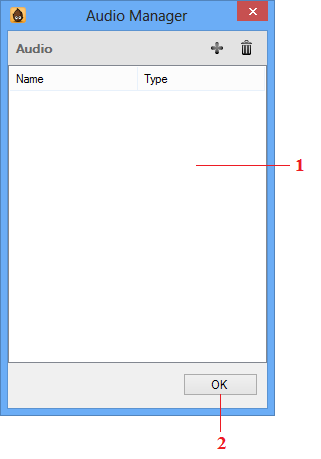


Hình 3.19 Màn hình Data Manger

|  |  |
| --- | --- |
| **STT** | **Ý nghĩa** |
| 1 | Hiển thị danh sách các thuộc tính đã thêm vào cho Data. Có 3 nút chức năng:   * Add: Thêm một thuộc tính vào danh sách các thuộc tính của Data. * Update: Cập nhật lại thông tin một thuộc tính đã thêm vào trong Data. * Delete: Xóa một thuộc tính đang được chọn trong danh sách các thuộc tính của Data. |
| 2 | Hiển thị thông tin của thuộc tính sẽ thêm vào cho Data. Các thông số bao gồm:   * Name: Tên thuộc tính sẽ thêm vào Data. * Type: Kiểu thuộc tính sẽ thêm vào Data. Chương trình hỗ trợ 3 kiểu thuộc tính: kiểu số, kiểu chữ, kiểu đúng sai. * Initial Value: Giá trị của thuộc tính. |
| 3 | Lưu thông tin của Data và tắt form, trở về form chính. |

Bảng 3.18 Bảng chú thích cho màn hình Data Manager

* Màn hình Audio Manager

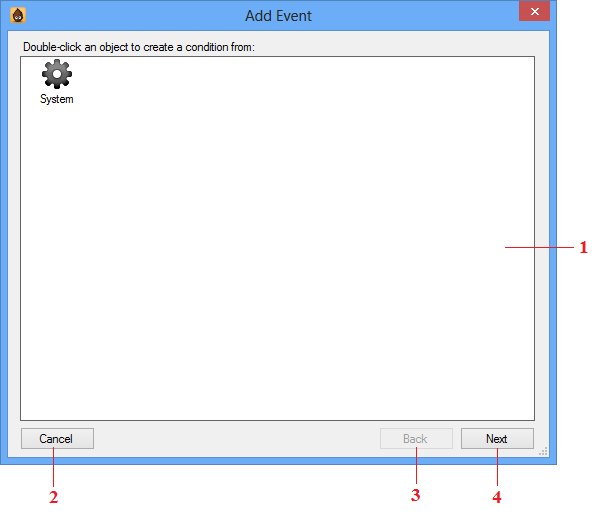


Hình 3.20 Màn hình Audio Manager

|  |  |
| --- | --- |
| **STT** | **Ý nghĩa** |
| 1 | Hiển thị danh sách các Audio đã thêm vào. Có 2 nút chức năng:   * Add: Thêm một Audio vào danh sách. * Delete: Xóa một Audio đang được chọn trong danh sách. |
| 2 | Lưu thông tin của Data và tắt form, trở về form chính. |

Bảng 3.19 Bảng chú thích cho màn hình Audio Manager

* Màn hình thêm Event

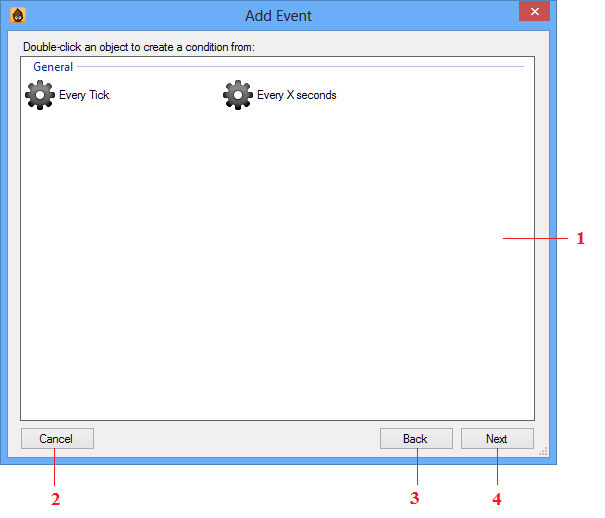


Hình 3.21 Màn hình thêm Event

|  |  |
| --- | --- |
| **STT** | **Ý nghĩa** |
| 1 | Hiển thị danh sách các đối tượng được thêm vào chương trình. |
| 2 | Không thêm sự kiện và trở về form chính. |
| 3 | Trở về bước trước đó. |
| 4 | Chọn đối tượng và tới bước kế tiếp. |

Bảng 3.20 Bảng chú thích cho màn hình thêm Event

* Màn hình thêm thông tin Event

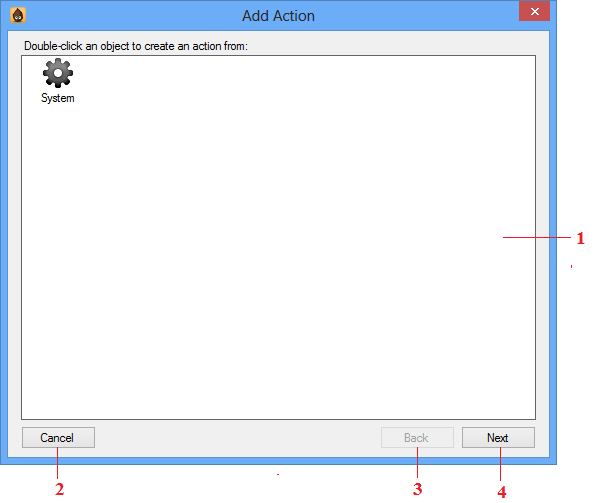


Hình 3.22 Màn hình thông tin Event

|  |  |
| --- | --- |
| **STT** | **Ý nghĩa** |
| 1 | Hiển thị danh sách các sự kiên mà đối tượng hỗ trợ. |
| 2 | Không thêm sự kiện và trở về form chính. |
| 3 | Trở về bước trước đó. |
| 4 | Chọn đối tượng và tới bước kế tiếp. |

Bảng 3.21 Bảng chú thích cho màn hình thông tin Event

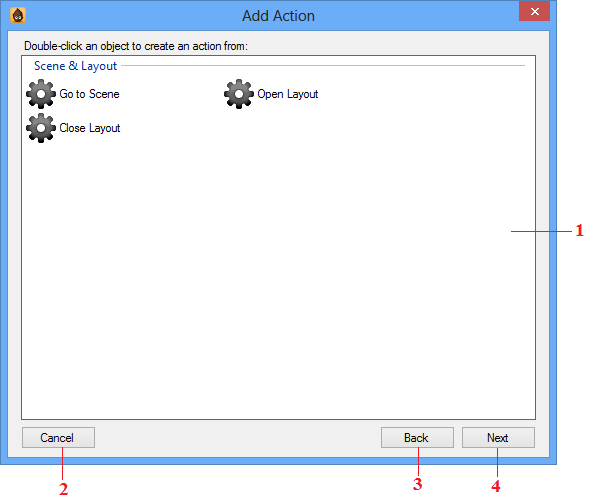
* Màn hình thêm Action

  
Hình 3.23 Màn hình Thêm Action

|  |  |
| --- | --- |
| **STT** | **Ý nghĩa** |
| 1 | Hiển thị danh sách các đối tượng được thêm vào chương trình. |
| 2 | Không thêm sự kiện và trở về form chính. |
| 3 | Trở về bước trước đó. |
| 4 | Chọn đối tượng và tới bước kế tiếp. |

Bảng 3.22 Bảng chú thích cho màn hình thêm Action

* Màn hình thêm thông tin Action



Hình 3.24 Màn hình thông tin Action

|  |  |
| --- | --- |
| **STT** | **Ý nghĩa** |
| 1 | Hiển thị danh sách các hành động mà đối tượng hỗ trợ. |
| 2 | Không thêm sự kiện và trở về form chính. |
| 3 | Trở về bước trước đó. |
| 4 | Chọn đối tượng và tới bước kế tiếp. |

Bảng 3.23 Bảng chú thích cho màn hình thông tin Action

CHƯƠNG 4: THIẾT KẾ FRAMEWORK EDESIGN

4.1 KHÁI QUÁT VỀ FRAMEWORK:

Framework eDesign được xây dựng và phát triển dựa trên nền tảng có sẵn của Cocos2d-x, với mục đích hỗ trợ cho công cụ hỗ trợ phát triển game EasyDesign và Cocos2d-x các gói tiện ích cơ bản. Với các thành phần tiêu biểu, *Behaviour* tạo các hành vi cho một đối tượng từ di chuyển, tính chất đến hiệu ứng. *ObjectManager* quản lý các đối tượng theo các phương thức được chọn như cây tứ phân, cây nhị phân, danh sách. *Map* thao tác với dữ liệu để tạo và lưu một bản đồ trong game. *Event* xây dựng các câu trúc sự kiện có sẵn để hỗ trợ cài đặt cho.

Nguyên nhân chính cần phải thiết kế framework eDesign:

* Tạo cơ chế tương tác trung gian giữa EasyDesign với Cocos2d-x
* Xây dựng hệ thống:
* Các hành vi (behaviours)
* Các hiệu ứng (Effect)
* Cơ chế quản lý các đối tượng (Cây tứ phân, cây nhị phân, danh sách)
* Hỗ trợ xây dựng khung diễn ảnh (animate)
* Xây dựng khung sự kiện (bàn phím, chuột, cảm ứng chạm, cảm ứng gia tốc)
* Hỗ trợ xây dựng cấu trúc bản đồ (map)
* Hỗ trợ xây dựng các dòng sự kiện

4.2 CÁC MÔ HÌNH GIỮA FRAMEWORK EDESIGN VÀ COCOS2D-X

Cocos2d-x vốn đã là một cấu trúc hỗ trợ đầy đủ cho quá trình xây dựng một hệ thống game. Với các lớp trung gian để có thể hoạt động mạnh mẽ trên nhiều môi trường khác nhau, Cocos2d-x có một cấu trúc đơn giản các đối tượng và phương thức. Vậy, Mô hình hệ thống game cơ bản được xây dựng bởi Cocos2d-x:

Hình 4.1 Mô hình hệ thống game cơ bản được xây dựng bởi Cocos2d-x

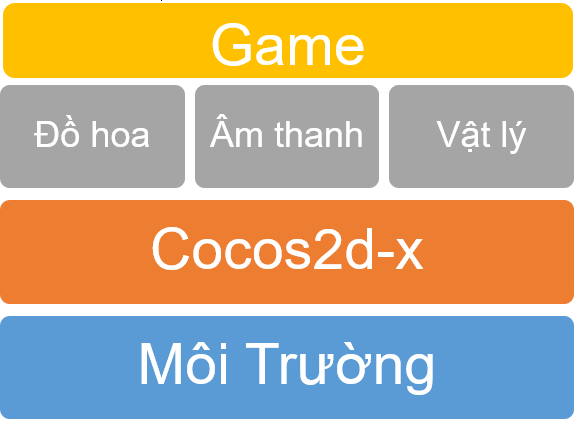
Dựa trên mô hình được vẽ có thể thấy các bước cơ bản để tạo cấu trúc game cơ bản của Cocos2d-x, Lớp Đại diện ứng dụng (App Delegate) sẽ tương tác để tạo các thông số cơ bản của game. Sau đó, Lớp đại diện ứng dụng sẽ gọi đến Lớp điều khiển (Director) để thực hiện lệnh chuyển đến cảnh (Scene) đầu tiên của game. Trước thao tác đó, một cảnh đã được tạo ra thông qua hàm tĩnh của một lớp (Layer) và cảnh đó được tạo ra để chứa lớp này.

Với kết quả thành công của quá trình tạo cảnh và lớp, lớp chính thức có thể thêm các nút (CCNode) vào để tạo thành một game, nhưng, trước tiên lớp phải thêm các yếu tố cơ bản như: *Sự kiện* để giúp cho lớp bắt được các hành động của người dùng theo ý muốn (chuột, bàn phím, cảm ứng chạm và cảm ứng gia tốc), *Âm thanh* thường được thêm vào lớp trong giai đoạn này (âm thanh có thể được thêm trước khi vào bên trong cảnh)

Các nút được thêm vào hoàn chỉnh có thể tạo thành một game cơ bản. Tuy nhiên, các nút này cần có có hành động (action) và thêm hiệu ứng (effect), diễn ảnh (animation) để làm cho game trung thực và tạo thêm nhiều ảnh hưởng đến cảm quan của người sử dụng.

Sau khi hoàn thành hết các hành động cần làm trong cảnh đó, Cảnh đó sẽ gọi lệnh để Lớp điều khiển (Director) chuyển sang cảnh khác và thực hiện lại các thao tác tạo lớp (layer), tạo nút (CCNode) và thêm hành động, hiệu ứng với diễn ảnh (animation) cho nút đó.

Mô hình cấu trúc Cocos2d-x:



Hình 4.2 Mô hình cấu trúc Cocos2d-x

Về tổng thể, Cocos2d-x đã hoàn toàn có thể tạo thành một game. Nhưng so về các API hỗ trợ, Cocos2d-x có nhiều hành động (action) nhưng lại thiếu các hành động tự phát sinh cho mỗi đối tượng nút hay còn gọi đúng hơn là các hành vi (behaviour) – như hành động tự động di chuyển về phía trước cho các nhân vật. Ngoài vấn đề trên, Cocos2d-x còn cần thêm các cấu trúc cây để quản lý các đối tượng trong game, giúp cho việc phát triển các dòng game có phạm vi thao tác rộng trở nên dễ dàng hơn.

Vậy, Cấu trúc khi được thêm Framework eDesign vào sẽ trở thành:

Hình 4.3 Mô hình cocos2d-x kết hợp framework eDesign

Với các yếu tố được thêm vào, Dòng tổ chức của một game được cấu trúc bởi Cocos2d-x hỗ trợ bằng Framework eDesign được hình thành với các bước cơ bản, Lớp đại diện (App Delegate) cấu trúc thông số, dùng lớp điều khiển (Director) để chuyển đến cảnh đầu tiên. Nhưng có một điểm khác với Cocos2d-x, framework hỗ trợ cho việc đọc tập dữ liệu có sẵn trong bản đồ (Map) để tạo thành các lớp (Layer) để chứa các đối tượng của game. Và, Các đối tượng của game (nút -Node) được quản lý bởi cấu trúc quản lý cây hoặc danh sách (List) để tiện cho việc thao tác vào cấu trúc. Một điểm đặc biệt khác so với phiên bản cấu trúc của Cocos2d-x, Các nút sẽ được thêm một yếu tố mới là hành vi để giúp tạo các hành động tự nhiên hơn. Như vậy các yếu tố được thêm trong khung Framework eDesign để hỗ trợ cho Cocos2d-x là:

* Bản đồ: để tạo cấu trúc lớp (layer) từ dữ liệu tập tin
* Quản lý đối tượng (ObjectManager): để quản lý các đối tượng nút được tạo ra trong quá trình tạo bản đồ hoặc tạo đối tượng của Lớp
* Hành vi (Behaviour): yếu tố được thêm vào để tạo các hành động tự nhiên hơn cho các nút

Ngoài các tính năng trên, Framework eDesign còn được thiết kế thêm các tính năng khác nhằm hỗ trợ phát huy tối đa khả năng của Cocos2d-x trên các nền tảng ngôn ngữ lập trình C++.

Mô hình chi tiết các tính năng được thiết kế trong Framework eDesign:

Hình 4.4 Sơ đồ tính năng của framework eDesign

Framework eDesign xây dựng trên các thành phần cơ bản của Cocos2d-x, do đó, các thành phần của eDesign cũng tận dụng tối đa các thành phần của Cocos2d-x bao gồm như:

* *Hành vi* (behaviour) được xây dựng trên hệ thống các hành động (action) có sẵn của Cocos2d-x.
* *Quản lý đối tượng* được cấu trúc dựa trên cơ chế nút (CCNode) và danh sách (list).
* *Diễn ảnh* dựa theo cơ chế diễn ảnh (animation) nhưng thêm các thành phần thích hợp để hỗ trợ phần diễn ảnh linh hoạt hơn.
* *Xử lý đầu vào,* tương tự như diễn ảnh, giúp cho quá trình cài đặt các sự kiện đầu vào đơn giản hơn và hỗ trợ cho quá trình thực hiện các sự kiện xử lý.
* *Bản đồ* hỗ trợ cho cơ chế bản đồ phân ô (TileMap) có sẵn của Cocos2d-x
* *Sự kiện xử lý* giúp quản lý các sự kiện trong game làm cho dòng sự kiện của game thống nhất
* *Hiệu ứng* hỗ trợ thêm một số hiệu ứng phức tạp và kế thừa các hiệu ứng (effect) sẵn có
* *Tương tác mạng xã hội*  hỗ trợ giao tiếp với các mạng xã hội thông qua ngôn ngữ C++

4.3 CÁC MÔ HÌNH GIỮA FRAMEWORK EDESIGN VÀ ỨNG DỤNG EASY DESIGN

Như đã được biết, Cocos2d-x vốn là một phương pháp (engine) đầy đủ để thiết kế một game, từ cơ bản đến phức tạp, cho cả 2D và 3D. Nhưng EasyDesign được thiết kế với cấu trúc tạo mã lệnh (code) còn hạn chế, nên quá trình giao tiếp giữa ứng dụng EasyDesign và Cocos2d-x gặp nhiều khó khắn. EasyDesign muốn thực hiện một thao tác cần phải thêm các mã lệnh vào nhiều vị trí khác nhau trong mã nguồn. Như vậy, rất khó để xác định vị trí thêm các đoạn mã thích hợp. Cocos2d-x đã hỗ trợ cơ chế làm gọn các mã lệnh nhưng không giải quyết được hoàn toàn vấn đề. Do vậy, việc hình thành lớp trung gian eDesign là một yếu tố cấp thiết để hỗ trợ cho EasyDesign được linh hoạt hơn, không cần phải tuân thủ hoàn toàn cấu trúc lệnh của cocos2d-x.

Mô hình hoạt động của ứng dụng EasyDesign để tạo ra mã nguồn cho game:

Hình 4.5 Mô hình xuất mã nguồn của easydesign

Với framework eDesign, ứng dụng EasyDesign có thể tự do thao tác với nền tảng .NET. Sau đó - khi xuất mã nguồn, ứng dụng EasyDesign chỉ cần thực hiện như việc xuất mã cho Cocos2d-x.Nhưng những phần khó khăn với ứng dụng, mà cần nhiều vị trí mã phối hợp với nhau, Framework eDesign sẽ giúp tối đơn giản vấn đề này. Và, làm cho một dòng mã có thể thay thế nhiều dòng mã khác nhất có thể.

Do vậy, quá trình chuyển đổi từ các thao tác kéo thả đơn giản trên nền của công cụ hỗ trợ phát triển game EasyDesign thành mã nguồn, sẽ được tối ưu hóa. Sao cho, mã nguồn được xuất ra là đơn giản nhất cho thao tác của EasyDesign và hỗ trợ tốt cho Cocos2d-x.

Mô hình hoạt động chuyển đổi từ thao tác trên EasyDesign chuyển sang mã nguồn trong Cocos2d-x:

Hình 4.6 Mô hình chuyển hoạt động từ easydesign sang cocos2d-x

Để hỗ trợ việc phân tách mã nguồn từ mã nguồn được xuất ra trong EasyDesign, framework eDesign hỗ trợ các tính năng:

Hình 4.7 Tính năng framework eDesign hỗ trợ ứng dụng EasyDesign

Mục đích thiết kế các tính năng: Hành vi, Diễn ảnh, Sự kiện đầu vào, Sự kiện xử lý, Tương tác mạng xã hội cho thích hợp với EasyDesign:

* *Hành vi:* giảm thiểu thao tác thiết kế nhưng chuỗi hành động (action) cho các đối tượng
* *Diễn ảnh:* Tương thích với cơ chế diễn ảnh của ứng dụng và đơn giản hơn các thao tác tạo ảnh động (animation) và chuyển thành cơ chế diễn ảnh (animate)
* *Sự kiện đầu vào:* Giúp cố định, tóm tắt các đoạn mã (code) để EasyDesign có thể dễ dàng xuất mã nguồn
* *Sự kiện xử lý:* thay vì chuyển các sự kiện xử lý đến từng đối tượng, framework sẽ hướng các sự kiện này theo nhóm để EasyDesign có thể định vị được vị trí in mã và xuất mã
* *Tương tác mạng xã hội:* Cơ chế tương tác mạng xã hội như facebook đã được hỗ trợ bởi nhiều ngôn ngữ và cộng cụ hỗ trợ phát triển game khác. Nhưng một cơ chế tương tác đa nền tảng mã nguồn mở để tiếp cận đến facebook qua ngôn ngữ C++ đang rất cần để nguyên cứu.

Quá trình trên, một phần chứng minh vai trò của framework eDesign với công cụ hỗ trợ phát triển game và cho cả Cocos2d-x điều là những phần cần thiết. Do đó, framework chú ý đến việc phát triển tiện lợi nhất thỏa mãn cho cả 2 mặt của vấn đề và góp phần đóng góp vào sự phát triển của phương pháp (engine) mã nguồn mở Cocos2d-x

4.4 THIẾT KẾ TÍNH NĂNG FRAMEWORK

Tính năng của framework eDesign được thiết kế từng mục giống với hình 16: “Sơ đồ tính năng của framework eDesign” bao gồm: hành vi (behavior), hiệu ứng (effect), danh sách (list), diễn ảnh (animate), sự kiện đầu vào (input), bản đồ (map), sự kiện xử lý (event), cây (tree), tương tác mạng xã hội (social network).

4.4.1 Tính năng Hành vi

Hành vi (behavior) là một chuỗi các hành động (action) hoặc các tính chất cơ bản của một nút đối tượng (Node). Hành vi được thiết kế để làm việc lập trình game trở nên trực quan hơn và còn làm nhẹ các chi phí lập trình cho những thao tác cơ bản vốn cần có của một đối tượng.

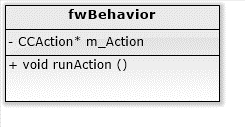
Hành vi được viết trên nền một lớp trừu tượng fwBehavior (kế thừa từ Node) với các Hành vi cơ bản:

Hình 4.8 Sơ đồ hành vi

4.4.1.1 Lớp fwBehavior

Lớp fwBehavior là lớp trựu tượng được các lớp hành vi khác kế thừa. Lớp fwBehavior được kế thừa từ lớp Node của Cocod2d-x nên các hàm đã có sẵn và có thể kế thừa trong lớp Node sẽ được bỏ qua trong quá trình vẽ các bản thiết kế. Ví dụ, hàm cập nhật (update), khởi tạo (init) và hàm tạo (create) có sẵn trong cấu trúc.

**Sơ đồ lớp fwBehavior:**



Hình 4.9 Sơ đồ lớp fwBehavior

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| CCAction\* | m\_Action | Sao chép các hành động của hành vi rồi gọi hàm thực thi trong đối tượng thực hiện hành vi |

Bảng 4.1 Thuộc tính lớp fwBehavior

**Phương thức:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| void | runAction | Không có | Gọi thực thi các hành động từ đối tượng thực hiện hành vi |

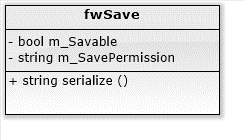
Bảng 4.2 Phương thức lớp fwBehavior

Lớp fwBehavior như một lớp trung gian giúp các lớp con được kế thứa có thể lưu trữ các hành động vừa được xử lý và là lớp được giao tiếp thực hiện hành động với các đối tượng thực hiện hành vi

4.4.1.2 Lớp fwSave

Lớp fwSave là lớp hành vi thuộc tính. Lớp này cho phép đối tượng thực thi cơ chế tự động lưu (Save) khi có lệnh gọi lưu bản đồ (save map). fwSave giải quyết cơ bản về cách lưu một đối tượng xuất tập tin cũng như là các thao tác xử lý dữ liệu trước khi lưu trữ.

**Sơ đồ lớp fwSave:**



Hình 4.10 Sơ đồ lớp fwSave

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| Bool | m\_Savable | Kiểm tra đối tượng có thỏa điều kiện lưu. Ví dụ, khởi tạo biến sẽ có giá trị đúng (trua) nhưng đối tượng đã được lưu và chưa có thay đổi từ lần lưu trước - biến sẽ có giá trị sai (false), hay đối tượng không cần phải lưu xuống nữa – biến cũng sẽ có giá trị sai (false) |
| String | m\_SavePermission | Mặc định đối tượng sẽ được sao lưu các dữ liệu cần thiết. Cấp phép có thể cho thêm các quyền lưu các giá trị ngoài hoặc không lưu một giá trị nào đó trong các mục mặc định |

Bảng 4.3 Thuộc tính lớp fwSave

**Phương thức:**

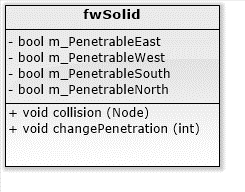
|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| String | serialize | Không có | Tạo cấu trúc lưu trữ cho đối tượng |

Bảng 4.4 Phương thức lớp fwSave

4.4.1.3 Lớp fwSolid

Lớp fwSolid là lớp hành vi thuộc tính. Lớp này cho phép đối tượng va chạm với tất cả các đối tượng khác với điều kiện không xuyên qua được. Tuy nhiên, với các điều kiện mong muốn có thể chỉnh thuộc tính của lớp fwSolid để một số đối tượng hoặc các đối tượng có thể xuyên qua được theo một hướng hay một điều kiện nhất định. Thường có đối tượng chứa lớp này trong game là Đất, Gạch, Đá, Vật cản .v.v...

**Sơ đồ lớp fwSolid**:



Hình 4.11 Sơ đồ lớp fwSolid

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| Bool | m\_ PenetrableEast | Kiểm tra có cho đi xuyên qua từ hướng đông |
| Bool | m\_PenetrableWest | Kiểm tra có cho đi xuyên qua từ hướng tây |
| Bool | m\_PenetrableSouth | Kiểm tra có cho đi xuyên qua từ hướng nam |
| Bool | m\_PenetrableNorth | Kiểm tra có cho đi xuyên qua từ hướng bắc |

Bảng 4.5 Thuộc tính lớp fwSolid

**Phương thức:**

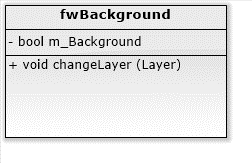
|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | Collision | Node | Kiểm tra các điều kiện va chạm và cập nhật tình trạng sau va chạm đối với các đối tượng va chạm |
| Void | changePenetration | Int | Dựa vào tham số truyền vào hàm tách tham số 4 chữ số đó thành các cờ để bật các kiểm tra cho đi xuyên qua. Ví dụ, 0001 cho phép đi qua hướng bắc và không cho đi xuyên các hướng còn lại |

Bảng 4.6 Phương thức lớp fwSolid

4.4.1.4 Lớp fwBackground

Lớp fwBackground là lớp hành vi thuộc tính đối lập với lớp fwSolid. Lớp này cho phép đối tượng được tách riêng vào một lớp khác (Layer), không cần thao tác xử lý va chạm với các đối tượng khác tại lớp (layer) xử lý các đối tượng chính trong game. Tuy nhiên, lớp fwBackground này có thể bị loại bỏ và đưa đối tượng vào lại lớp chính. Ví dụ, Núi, Cây, Cỏ, Mặt trời, lâu đài ẩn, cửa ngầm .v..v...

**Sơ đồ lớp fwBackground:**



Hình 4.12 Sơ đồ lớp fwBackground

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| Bool | m\_Background | Kiểm tra đối tượng có còn thuộc lớp Background |

Bảng 4.7 Thuộc tính lớp fwBackground

**Phương thức:**

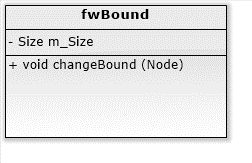
|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | changeLayer | Layer | Đổi đối tượng từ lớp Background sang lớp xử lý va chạm cho các đối tượng chính |

Bảng 4.8 Phương thức lớp fwBackground

4.4.1.5 Lớp fwBound

Lớp fwBound là lớp hành vi thuộc tính. Lớp này cho phép đối tượng di chuyển đóng khung trong một đối tượng hoặc phạm vi quy định. Tuy nhiên, Lớp có hỗ trợ hàm để chuyển đổi khung bao bọc của đối tượng. Ví dụ, Khung di chuyển của quái, khung di chuyển cho người chơi, khung di chuyển do vật phẩm .v..v...

**Sơ đồ lớp fwBound:**



Hình 4.13 Sơ đồ lớp fwBound

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| Size | m\_Size | Kích thước vùng bao bọc đối tượng |

Bảng 4.9 Thuộc tính lớp fwBound

**Phương thức:**

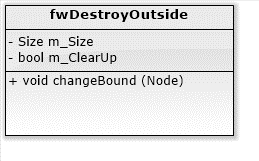
|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | changeBound | Node | Thay đổi vùng bao bọc cho đối tượng, riêng chỉ thay đổi Size đơn thuần thôi, người dùng có thể sử dụng hàm SetSize có sẵn của lớp |

Bảng 4.10 Phương thức lớp fwBound

4.4.1.6 Lớp fwDestroyOutside

Lớp fwDestroyOutside là lớp hành vi thuộc tính. Lớp này cho phép đối tượng bị hủy khi ra khỏi vùng bao bọc. Tuy nhiên, Lớp có hỗ trợ thao tác thay đổi phạm vi hủy đối tượng và hỗ trợ thêm cơ chế xóa hẳn đối tượng ra khỏi đối tượng chứa (parent node). Ví dụ, đạn khi ra khỏi màn hình, vật phẩm khi ra khỏi cảnh .v..v...

**Sơ đồ lớp fwDestroyOutside:**



Hình 4.14 Sơ đồ lớp fwDestrouOutside

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| Size | m\_Size | Kích thước vùng bao an toàn của đối tượng. Vùng an toàn là pham vị đối tượng không bị hủy nhưng khi ra khỏi phạm vi này đối tượng sẽ bị hủy |
| Bool | m\_ClearUp | Hỗ trợ cơ chế xóa hoàn toàn đối tượng khỏi đối tượng chứa |

Bảng 4.11 Thuộc tính lớp fwDestroyOutside

**Phương thức:**

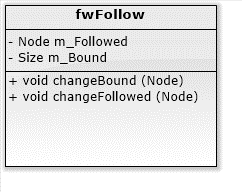
|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | changeBound | Node | Thay đổi vùng an toàn cho đối tượng, riêng chỉ thay đổi Size đơn thuần thôi, người dùng có thể sử dụng hàm SetSize có sẵn của lớp |

Bảng 4.12 Phương thức lớp fwDestroyOutside

4.4.1.7 Lớp fwFollow

Lớp fwFollow là lớp hành vi hành động. Lớp này cho phép đối tượng theo sau một đối tượng khác, bao gồm việc cho máy quay (camera) chạy theo đối tượng chính của cảnh game. Lớp còn hỗ trợ thao tác thay đổi đối tượng bị theo sau. Ví dụ, Tên nhân vật chạy theo đối tượng, Ống máu chạy theo đối tượng, Tên lửa định vị

**Sơ đồ lớp fwFollow:**



Hình 4.15 Sơ đồ lớp fwFollow

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| Node | m\_Followed | Đối tượng được theo sau |
| Size | m\_Bound | Kích thước vùng theo dõi của đối tượng. Nếu đối tượng bị theo sau vượt quá phạm vị của vùng theo dõi, Đối tượng chính sẽ không tiếp tục theo sau nữa |

Bảng 4.13 Thuộc tính lớp fwFollow

**Phương thức:**

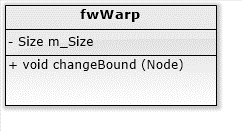
|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | changeBound | Node | Thay đổi vùng theo dõi cho đối tượng, riêng chỉ thay đổi Size đơn thuần thôi, người dùng có thể sử dụng hàm SetBound có sẵn của lớp |
| Void | changeFollowed | Node | Thay đổi đối tượng được theo dõi |

Bảng 4.14 Phương thức lớp fwFollow

4.4.1.8 Lớp fwWarp

Lớp fwWarp là một lớp hành vi hành động. Lớp này cho phép đối tượng chuyển từ phía bên này của vùng bao di chuyển đến phía bên kia. Tuy nhiên, Lớp có thao tác hỗ trợ thay đổi phạm vi vùng bao. Ví dụ, Các loại đạn đặc biệt, các loại quái vật đặc biệt, trùm .v..v...

**Sơ đồ lớp fwWarp:**



Hình 4.16 Sơ đồ lớp fwWarp

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| Size | m\_Size | Kích thước phạm vi đối tượng không bị chuyển vị trí |

Bảng 4.15 Thuộc tính lớp fwWarp

**Phương thức:**

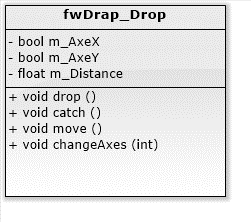
|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | changeBound | Node | Thay đổi vùng phạm vi đối tượng không bị chuyển vị trí, riêng chỉ thay đổi Size đơn thuần thôi, người dùng có thể sử dụng hàm SetSize có sẵn của lớp |

Bảng 4.16 Phương thức lớp fwWarp

4.4.1.9 Lớp fwDrap\_Drop

Lớp fwDrap\_Drop là một lớp hành vi hành động. Lớp nào cho phép đối tượng được nắm kéo và thả đi bằng chuột hoặc cảm ứng chạm của di động hay máy tính bảng. Lớp còn thiết kế thêm thao tác thả đối tượng khi gặp điều kiện yêu cầu. Ví dụ, kéo và chỉnh góc bắn cho chim trong Angry Bird, kéo và thả các đối tượng trong game thế loại xây tháp .v..v...

**Sơ đồ lớp fwDrap\_Drop:**



Hình 4.17 Sơ đồ lớp fwDrap\_Drop

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| bool | m\_AxeX | Cho phép đối tượng di chuyển theo trục X |
| bool | m\_AxeY | Cho phép đối tượng di chuyển theo trục Y |
| float | m\_Distance | Khoảng cách (độ chính xác) để nắm kéo (drap) đối tượng |

Bảng 4.17 Thuộc tính lớp fwDrap\_Drop

**Phương thức:**

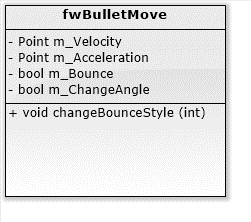
|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | drop | Không có | Thả ngay đối tượng xuống |
| Void | catch | Không có | Bắt đầu nắm kéo đối tượng |
| Void | move | Không có | Di chuyển đối tượng |
| Void | changeAxes | int | Tham số truyền vào là một kiểu số nguyên có 2 chữ số giúp định vị đối tượng có thể di chuyển theo trục nào. Ví dụ, 11 là đối tượng có thể di chuyển theo cả 2 trục |

Bảng 4.18 Phương thức lớp fwDrap\_Drop

4.4.1.10 Lớp fwBulletMove

Lớp fwBulletMove là một lớp hành vi hành động. Lớp này cho phép đối tượng di chuyển theo các phương và hướng đã định với quỹ đạo như đường đi viên đạn. Lớp này còn có các phương thức tinh chỉnh để đối tượng có thể di chuyển qua hoặc bị cản bởi một đối tượng vật cứng (fwSolid). Ví dụ, Đạn bắn, quái vật di chuyển, vật phẩm trượt trên nền đất .v..v...

**Sơ đồ lớp fwBulletMove:**



Hình 4.18 Sơ đồ lớp fwBulletMove

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| Point | m\_Velocity | Vận tốc của đối tượng theo 2 trục X, Y |
| Point | m\_Acceleration | Gia tốc của đối tượng theo 2 trục X, Y |
| Bool | m\_Bounce | Kiểm tra có dội khi gặp đối tượng cứng (fwSolid) |
| Bool | m\_ChangeAngle | Kiểm tra có theo đổi góc của đối tượng khi va chạm |

Bảng 4.19 Thuộc tính lớp fwBulletMove

**Phương thức:**

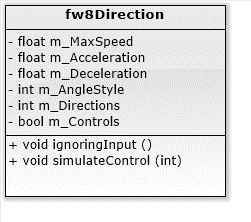
|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | changeBounceStyle | int | Tham số truyền vào là loại va chạm mà đối tượng sử thực hiện khi va chạm với đối tượng cứng. Có các loại cơ bản như: đội lại (Deflecting), Đẩy (Push), Trượt (Slide), dính chặt |

Bảng 4.20 Phương thức lớp fwBulletMove

4.4.1.11 Lớp fw8Direction

Lớp fw8Direction là một lớp hành vi hành động. Lớp này cho phép đối tượng được điều khiển bởi bàn phím có thể di chuyển theo tám hướng khác nhau: trước, sau, trên dưới, trước trên, trước dưới, sau trên, sau dưới hoặc có thể đơn giản chỉ 4 hướng cơ bản. Lớp này còn thêm các thao tác giúp cho việc chuyển hướng nhiều nhiều cơ chế. Ví dụ, game có góc nhìn trên xuống – tank – bomber man, điều kiển đi chuyển trong môi trường nước – mario cảnh dưới nước .v..v...

**Sơ đồ lớp fw8Direction:**



Hình 4.19 Sơ đồ lớp fw8Direction

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| float | m\_MaxSpeed | Vận tốc tối đa đối tượng đạt được |
| float | m\_Acceleration | Gia tốc của đối tượng |
| float | m\_Deceleration | Gia tốc hãm của đối tượng |
| int | m\_AngleStyle | Cách thay đổi góc di chuyển của đối tượng. Có 4 cách cơ bản: 45 độ, 90 độ, 360 độ và không thay đổi góc |
| int | m\_Directions | Cách xử lý di chuyển cho đối tượng. Có 4 cách cơ bản: 2 hướng (trên – dưới và trái – phải) 4 hướng, 8 hướng |
| bool | m\_Controls | Đối tượng có điều khiển ngay đầu game |

Bảng 4.21 Thuộc tính lớp fw8Direction

**Phương thức:**

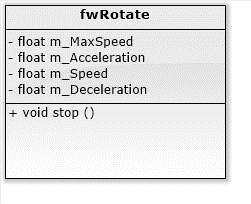
|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | ignoringInput | Không có | Loại bỏ các dữ liệu đầu vào |
| void | simulateControl | int | Giả lập như đang được điều khiển bởi sự kiện đầu vào với tham số (int) như là hướng đang được điều khiển |

Bảng 4.22 Phương thức lớp fw8Direction

4.4.1.12 Lớp fwRotate

Lớp fwRotate là một lớp hành vi hành động. Lớp này cho phép đối tượng chuyển động tròn xung quanh trục cố định. Lớp này cơ bản để xử lý các chuyển động vừa xoay vừa di chuyển hoặc xử lý các đối tượng xoay quanh trục. Ví dụ, đạn trong contra, các thanh xoay trong mario .v..v...

**Sơ đồ lớp fwRotate:**



Hình 4.20 Sơ đồ lớp fwRotate

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| float | m\_MaxSpeed | Vận tốc tối đa đối tượng đạt được |
| float | m\_Acceleration | Gia tốc của đối tượng |
| float | m\_Speed | Vận tốc khởi điểm của đối tượng |
| float | m\_Deceleration | Gia tốc hãm của đối tượng |

Bảng 4.23 Thuộc tính lớp fwRotate

**Phương thức:**

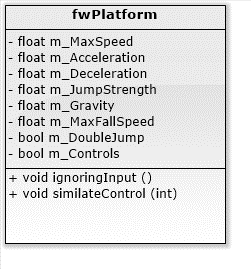
|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | stop | Không có | Ngừng chuyển động của đối tượng |

Bảng 4.24 Phương thức lớp fwRotate

4.4.1.13 Lớp fwPlatform

Lớp fwPlatform là một lớp hành vi hành động. Lớp này cho phép đối tượng được điều khiển bởi dữ liệu đầu vào từ bàn phím như di chuyển về phía trước, lùi lại, nhảy lên và chui qua. Đặc biệt, Lớp hành vi này sẽ tự động xét va chạm và xem như mình bị cản đối với các đối tượng cứng (fwSolid). Lớp hành vi này thường dùng cho game 2D màn hình ngang như Mario, Contra, Megaman .v..v...

**Sơ đồ lớp fwPlatform:**



Hình 4.21 Sơ đồ lớp fwPlatform

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| float | m\_MaxSpeed | Vận tốc tối đa đối tượng đạt được |
| float | m\_Acceleration | Gia tốc của đối tượng |
| float | m\_Deceleration | Gia tốc hãm của đối tượng |
| float | m\_JumpStrength | Sức bật của đối tượng |
| float | m\_Gravity | Trọng lực của đối tượng |
| float | m\_MaxFallSpeed | Tốc độ rơi tối đa của đối tượng |
| bool | m\_DoubleJump | Kiểm tra đối tượng có khả năng nhảy 2 lần |
| bool | m\_Controls | Đối tượng có điều khiển ngay đầu game |

Bảng 4.25 Thuộc tính lớp fwPlatform

**Phương thức:**

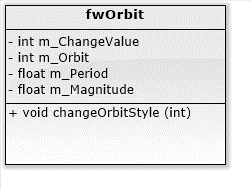
|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | ignoringInput | Không có | Loại bỏ các dữ liệu đầu vào |
| void | simulateControl | int | Giả lập như đang được điều khiển bởi sự kiện đầu vào với tham số (int) như là thao tác đang điều khiển |

Bảng 4.26 Phương thức lớp fwPlatform

4.4.1.14 Lớp fwOrbit

Lớp fwOrbit là một lớp hành vi hành động. Lớp này cho phép đối tượng di chuyển, thay đổi kích thước, giá trị theo quỹ đạo hình sin, dạng lưỡi cưa. Lớp này thường dùng cho các hiệu ứng thay đổi có quỹ đạo. Ví dụ, cách di chuyển của các thanh nhảy trong mario, các vật cản .v..v...

**Sơ đồ lớp fwOrbit:**



Hình 4.22 Sơ đồ lớp fwOrbit

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| int | m\_ChangeValue | Giá trị của đối tượng sẽ được thay đổi theo quỹ đạo. Ví dụ, di chuyển, kích thước, thuộc tính |
| int | m\_Orbit | Loại quỹ đạo để thay đổi giá trị |
| float | m\_Period | Thời gian cho 1 chu kỳ của quỹ đạo |
| float | m\_Magnitude | Phạm vi thay đổi giá trị của một chu kỳ |

Bảng 4.27 Thuộc tính lớp fwOrbit

**Phương thức:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | changeOrbitStyle | int | Tham số truyền vào là loại quỹ đạo sẽ được chuyển. Có các loại quỹ đạo như: hình sin, lưỡi cưa, đồng hồ, dao động điều hòa |

Bảng 4.28 Phương thức lớp fwOrbit

4.4.2 Tính năng Hiệu ứng

Hiệu ứng (Effect) là tính năng giúp hỗ trợ kỹ thuật hiển thị cho các nút đối tượng (Node). Ngoài ra, Hiệu ứng còn cung cấp các tính năng tùy chỉnh giúp việc chuyển đổi hiệu ứng trong một nút đối tượng trở nên linh hoạt hơn và tiện cho việc thiết kế game trực quan hơn.

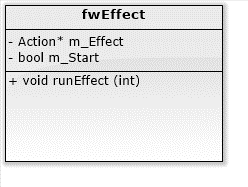
Hiệu ứng (Effect) được thiết kế trên nền lớp fwEffect (kế thừa từ Node) với các hiệu ứng cơ bản

Hình 4.23 Sơ đồ hiệu ứng

4.4.2.1 Lớp fwEffect

Lớp fwEffect là lớp trựu tượng được các lớp hiệu ứng khác kế thừa. Lớp fwEffect được kế thừa từ lớp Node của Cocod2d-x nên các hàm đã có sẵn và có thể kế thừa trong lớp Node sẽ được bỏ qua trong quá trình vẽ các bản thiết kế. Ví dụ, hàm cập nhật (update), khởi tạo (init) và hàm tạo (create) có sẵn trong cấu trúc.

**Sơ đồ lớp fwEffect:**



Hình 4.24 Sơ đồ lớp fwEffect

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| Action\* | m\_Effect | Sao chép các hiệu ứng rồi gọi hàm thực thi trong đối tượng hiện thực hiệu ứng |
| bool | m\_Start | Kiểm tra hiệu ứng có thực hiện ngay từ đầu |

Bảng 4.29 Thuộc tính lớp fwEffect

**Phương thức:**

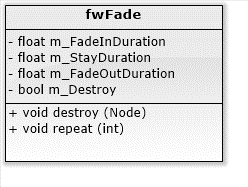
|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | runEffect | Không có | Gọi lệnh thực thi các hiệu ứng từ các đối tượng chứa hiệu ứng (Parent Node) |

Bảng 4.30 Phương thức lớp fwEffect

4.4.2.2 Lớp fwFade

Lớp fwFade là một lớp hiệu ứng hình ảnh. Lớp này cho phép đối tượng thực hiện hiệu ứng hiện dần lên ( FadeIn) và phai dần đi (FadeOut). Lớp thêm các hàm hỗ trợ thiết lập thời gian hiện dần lên, đợi đến khi mất, phai dần đi, hủy đối tượng khi đối tượng phai đi và thiết lập chế độ vĩnh viễn. Ví dụ, Hiệu ứng nổ, các vật phẩm có thời gian .v..v...

**Sơ đồ lớp fwFade:**



Hình 4.25 Sơ đồ lớp fwFade

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| float | m\_FadeInDuration | Thời gian đối tượng hiện dần lên |
| float | m\_StayDuration | Thời gian đối tượng hiện hoàn toàn trên màn hình |
| float | m\_FadeOutDuration | Thời gian đối tượng phai dần đi |
| Bool | m\_Destroy | Kiểm tra đối tượng có cần xóa khi ẩn hoàn toàn |

Bảng 4.31 Thuộc tính lớp fwFade

**Phương thức:**

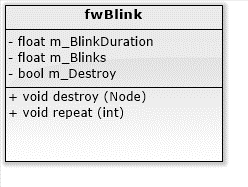
|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | destroy | Node | Dùng hàm chọn lọc (selector) để hướng đối tượng đến lệnh kiểm tra có hủy đối tượng khi ẩn |
| Void | repeat | Int | Nếu tham số truyền vào là số dương không âm (> 0) sẽ thực hiện hiệu ứng tham số lần trước khi ẩn hẳn. Nếu truyền số âm (< 0) hiệu ứng sẽ lập vĩnh viễn |

Bảng 4.32 Phương thức lớp fwFade

4.4.2.3 Lớp fwBlink

Lớp fwBlink là một lớp hiệu ứng hình ảnh. Lớp này cho phép đối tượng thực thi hiệu ứng chớp. Tương đối khác với phai (Fade), Chớp là hiệu ứng đối tượng đang hiện chuyển sang trạng thái ẩn và rồi hiện lên lại. Lớp còn hỗ trợ thêm cơ chế chớp vĩnh viễn, ngừng chớp và thay đổi thông số chớp qua các sự kiện. Ví dụ, Tia sét, Quái vật, Đạn .v..v...

**Sơ đồ lớp fwBlink:**



Hình 4.26 Sơ đồ lớp fwBlink

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| float | m\_BlinkDuration | Thời gian đối tượng thực hiện chớp |
| int | m\_Blinks | Số lần đối tượng chớp trong thời gian thực hiện |
| Bool | m\_Destroy | Kiểm tra đối tượng có cần xóa khi chớp hoàn toàn |

Bảng 4.34 Thuộc tính lớp fwBlink

**Phương thức:**

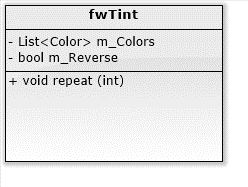
|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | destroy | Node | Dùng hàm chọn lọc (selector) để hướng đối tượng đến lệnh kiểm tra có hủy đối tượng khi chớp hoàn thành |
| Void | repeat | Int | Nếu tham số truyền vào là số dương không âm (> 0) sẽ thực hiện hiệu ứng tham số lần trước khi ngừng hẳn. Nếu truyền số âm (< 0) hiệu ứng sẽ lập vĩnh viễn |

Bảng 4.34 Phương thức lớp fwBlink

4.4.2.4 Lớp fwTint

Lớp fwTint là một lớp hiệu ứng hình ảnh. Lớp này cho phép đối tượng có thể chuyển đổi màu sắc từ màu này sang màu khác và chuyển ngược lại. Lớp còn cấu hình thêm để đối tượng có thể chuyển nhiều màu liên tục, và sau đó, có thể tinh chỉnh để chuyển ngược lại. Ví dụ, lúc nhân vật biến hình hoặc chuyển trạng thái, lúc thu thập các vật phẩm đặc biệt .v..v...

**Sơ đồ lớp fwTint:**



Hình 4.27 Sơ đồ lớp fwTint

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| float | m\_BlinkDuration | Thời gian đối tượng thực hiện chớp |
| int | m\_Blinks | Số lần đối tượng chớp trong thời gian thực hiện |
| Bool | m\_Destroy | Kiểm tra đối tượng có cần xóa khi chớp hoàn toàn |

Bảng 4.35 Thuộc tính lớp fwTint

**Phương thức:**

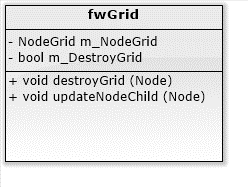
|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | repeat | Int | Nếu tham số truyền vào là số dương không âm (> 0) sẽ thực hiện hiệu ứng tham số lần trước khi ngừng hẳn. Nếu truyền số âm (< 0) hiệu ứng sẽ lập vĩnh viễn |

Bảng 4.36 Phương thức lớp Tint

4.4.2.5 Lớp fwGrid

Lớp fwGrid là một lớp hiệu ứng hình ảnh. Lớp này cho phép đối tượng tạo các hiệu ứng hình ảnh dạng lưới. Riêng, các hiệu ứng của lới đã được hỗ trợ bởi Cocos2d-x nên framework eDesign chỉ hỗ trợ phần tạo lưới để thực thi các hiệu ứng lưới. Ví dụ, mắt cá (lens), chất lỏng (liquid), rợn sóng (wave), tự trộn (shuffle) .v..v...

**Sơ đồ lớp fwGrid:**



Hình 4.28 Sơ đồ lớp fwGrid

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| NodeGrid | m\_NodeGrid | Lưới chứa các đối tượng thực hiện hiệu ứng |
| Bool | m\_DestroyGrid | Kiểm tra quá trình trả đối tượng trong lưới về lớp chính |

Bảng 4.37 Thuộc tính lớp fwGrid

**Phương thức:**

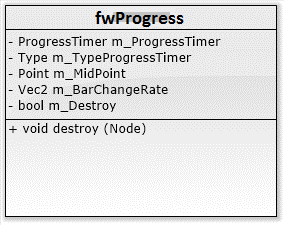
|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | destroyGrid | Node | Dùng hàm chọn lọc (selector) để hướng đối tượng đến lệnh kiểm tra có hủy lưới và trả các đối tượng bên trong lưới về với lớp chính |
| Void | updateNodeChild | Node | Tham số truyền vào là đối tượng cần thêm vào lưới, có thể là một lớp (layer) hay chỉ là một đối tượng đã được thêm vào lớp (sprite, label) |

Bảng 4.38 Phương thức lớp fwGrid

4.4.2.6 Lớp fwProgress

Lớp fwProgress là một lớp hiệu ứng hình ảnh. Lớp này cho phép đối tượng hiển thị ra theo các định dạng nhất định. Lớp còn hỗ trợ các tùy chỉnh như chỉ hiện thao chiều dọc, chỉ hiển thị chiều ngang và tùy chỉnh giá trị hiển thị. Ví dụ, Quái vật xuất hiện từ dưới lên, khi tiêu diệt xong trùm, trùm bị tiêu diệt.

**Sơ đồ lớp fwProgress:**



Hình 4.29 Sơ đồ lớp fwProgress

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| ProgressTimer | m\_ProgressTimer | Đối tượng tiến trình thời gian để thực thi tiến trình hiển thị của đối tượng |
| Type | m\_TypeProgressTimer | Loại hiển thị của tiến trình thời gian. Có 2 loại cơ bản: thanh (BAR), ra đa (radial) |
| Point | m\_MidPoint | Điểm bắt đầu hiển thị hiệu ứng. Hiệu ứng hiển thị sẽ từ điểm này mà đi ra 2 bên cho đến khi nào đụng biên hiển thị (không hiển thị thêm được) |
| Vec2 | m\_BarChangeRate | Tỉ lệ thay đổi với 2 giá trị x,y. Ví dụ, nếu x = 0 tức là không cần hiển thị thêm theo trục x. Giá trị có thể từ 0 -> n. Nếu giá trị lớn hơn 1 (100%) thì đối tượng sẽ đc hiển thị sẵn n \* 100% sau đó thu về 0% và tăng lên lại. |
| bool | m\_Destroy | Kiểm tra xóa đối tượng khi hoàn thành hiệu ứng |

Bảng 4.39 Thuộc tính lớp fwProgress

**Phương thức:**

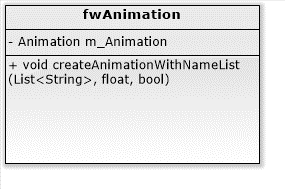
|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | destroy | Node | Dùng hàm chọn lọc (selector) để hướng đối tượng đến lệnh kiểm tra có hủy đối tượng khi hiệu ứng hoàn thành |

Bảng 4.40 Phương thức lớp fwProgress

4.4.3 Tính năng Diễn ảnh

Tính năng Diễn ảnh (animate) được đưa ra để hỗ trợ đơn giản mã nguồn khi công cụ hỗ trợ phát triển game EasyDesign cần tạo ra một ảnh động (Animation). Tính năng Diễn ảnh về cơ bản đã được hỗ trợ sẵn bởi Cocos2d-x. Nhưng, tính năng này được framework hỗ trợ thêm phần tạo diễn ảnh bằng danh sách tên các khung hình.

**Sơ đồ lớp fwAnimation:**



Hình 4.30 Sơ đồ lớp fwAnimation

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| Animation | m\_Animation | Ảnh động sao khi được khởi tạo |

Bảng 4.41 Thuộc tính lớp fwAnimation

**Phương thức:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | createAnimationWithNameList | List<string> tham số 1, float tham số 2, bool tham số 3 | Tham số 1 sẽ truyền vào tên các hình sẽ được tạo thành ảnh động, tham số 2 truyền vào thời gian ngừng giữa 2 lần chuyển ảnh, tham số 3 tinh chỉnh xem khi lập lại ảnh động có quay về khung hình chính |

Bảng 4.42 Phương thức lớp fwAnimation

Ngoài cách sử dụng tính năng fwAnimation, công cụ hỗ trợ phát triển game EasyDesign còn hỗ trợ cho dùng các lệnh có sẵn của Cocos2d-x để tạo ảnh động cho đối tượng nên tính năng fwAnimation chỉ đơn giản hỗ trợ một phần cho EasyDesign trong thao tác này.

4.4.4 Tính năng Xử lý đầu vào

Xử lý đầu vào (input) giúp xử lý thông tin đầu vào từ các nguồn như: Bàn phím, Chuột, Chạm màn hình cảm ứng chạm, cảm ứng gia tốc. Các lớp này đã được hỗ trợ bởi Cocos2d-x nhưng chỉ đơn thuần là các thông tin cơ bản từ các nguồn vào. Ví dụ, đối với bàn phím cocos2d-x chỉ hỗ bắt các phím vừa được nhấn và thả, nhưng, lại không hỗ trợ các thao tác đối với các phím đang được giữ lại .v..v...

Ngoài ra, tính năng xử lý đầu vào mục đích thiết kế là hỗ trợ cho EasyDesign trong việc thiết kế mã nguồn.

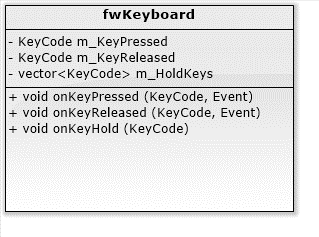
Hình 4.31 Sơ đồ tính năng xử lý đầu vào

4.4.4.1 Bàn phím

Bàn phím là xử lý đầu vào chung nhất dành cho các đối tượng. Các đối tượng điều có thể nhận và chạy dữ liệu đã được xử lý từ bàn phím. Riêng các hành vi có sẵn bắt sự kiện từ bàn phím sẽ được cài đặt sẵn các bắt sự kiện vào bên trong. Nên, xử lý đầu vào bàn phím dành cho các đối tượng ngoài mục hành vi đó.

Lớp được thiết kế singleton, nhưng, do cấu trúc singleton là thống nhất với mọi lớp nên sẽ không trình bài trong phần sơ đồ lớp.

**Sơ đồ lớp fwKeyboard:**



Hình 4.32 Sơ đồ lớp fwKeybroad

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| KeyCode | m\_KeyPressed | Phím vừa được ấn |
| KeyCode | m\_KeyReleased | Phím vừa được thả |
| Vector<KeyCode> | m\_HoldKeys | Những phím đang được giữ |

Bảng 4.43 Thuộc tính lớp fwKeybroad

**Phương thức:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | onKeyPressed | KeyCode – Tham số 1, Event – Tham số 2 | Tham số 1 truyền vào phím được ấn, event là đối tượng sự kiện trả về |
| Void | onKeyReleased | KeyCode – Tham số 1, Event – Tham số 2 | Tham số 1 truyền vào phím được thả, event là đối tượng sự kiện trả về |
| void | onKeyHold | KeyCode | Tham số truyền vào là phím muốn được kiểm tra có đang được giữ |

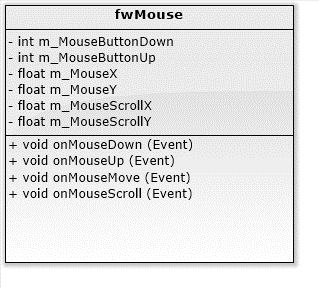
Bảng 4.44 Phương thức lớp fwKeybroad

4.4.4.2 Chuột

Chuột là xử lý đầu vào quan trọng trên thiết bị máy tính để bàn (desktop). Do đó, cocos2d-x đã hỗ trợ gần như hoàn thiện. Nhưng với hầu hết các loại game, ngoài việc bắt sự kiện cho từng đối tượng thì bắt sự kiện và truy xuất đến giá trị của tín hiệu đầu vào cũng có phần quan trọng. Ví dụ, game thể loại FPS, chuột thường được xem như là vị trí ngắm đạn, do đó việc nhiều đối tượng cần bắt sự kiện chung có tín hiệu đầu vào ở chuột là rất cần thiết.

Do vậy, framework eDesign đơn giản hỗ trợ bắt tín hiệu đầu vào và tạo biến thể singleton để mọi đối tượng có thể truy xuất tọa độ chuột. Việc này góp phần hỗ trợ cho công cụ hỗ trợ phát triển game EasyDesign các bước cơ bản. Do cấu trúc singleton là thống nhất với mọi lớp nên sẽ không trình bài trong phần sơ đồ lớp.

**Sơ đồ lớp fwMouse:**



Hình 4.33 Sơ đồ lớp fwMouse

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| int | m\_MouseButtonDown | Nút vừa được ấn trên chuột |
| int | m\_MouseButtonUp | Nút vừa được thả trên chuột |
| float | m\_MouseX | Tọa độ trục hoành của chuột |
| float | m\_MouseY | Tọa độ trục tung của chuột |
| float | m\_MouseScrollX | Giá trị nút cuộn trả về theo trục hoành |
| float | m\_MouseScrollY | Giá trị nút cuộn trả về theo trục tung |

Bảng 4.45 Thuộc tính lớp fwMouse

**Phương thức:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | onMouseDown | Event | Tham số trả về một sự kiện chuột (EventMouse) để lấy giá trị nút vừa ấn |
| Void | onMouseUp | Event | Tham số trả về một sự kiện chuột (EventMouse) để lấy giá trị nút vừa thả |
| Void | onMouseMove | Event | Tham số trả về một sự kiện chuột (EventMouse) để lấy tọa độ chuột trên màn hình hoặc lớp (layer) |
| Void | onMouseScroll | Event | Tham số trả về một sự kiện chuột (EventMouse) để lấy giá trị cuộn chuột theo trục tung và trục hoành |

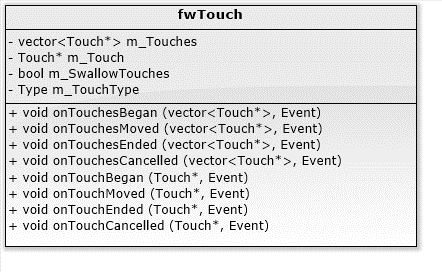
Bảng 4.46 Phương thức lớp fwMouse

4.4.4.3 Cảm ứng chạm

Hơn cả sức quan trọng của Chuột với máy tính để bàn, cảm ứng chạm gần như là một xử lý đầu vào (input) không thể thiếu đối với điện thoại di động. Hầu hết các thể loại game trên di động điều phải xử lý tín hiệu đầu vào này. Tuy nhiên cũng như vấn đề của xử lý tín hiệu đầu vào của chuột, Tín hiệu xử lý đầu vào của cảm ứng chạm được Cocos2d-x xử lý sự kiện riêng cho từng đối tượng. Vì vậy, các đối tượng khác không thể quan sát hay truy xuất dữ liệu từ cảm ứng chạm. Chính việc này dẫn đến cấu trúc của game trở nên phức tạp nhưng rất tỉ mỉ.

Công cụ hỗ trợ phát triển game EasyDesign gặp nhiều khó khăn khi tiếp cận với cấu trúc tỉ mỉ như vậy. Framework eDesign đã hỗ trợ EasyDesign với cấu trúc lớp fwTouch, có thiết kế singleton, để đơn giản hóa một số vấn đề. Riêng các đối tượng đã có trực tiếp nhận dữ liệu xử lý cảm ứng chạm sẽ không bị ảnh hưởng. Ví dụ, fwDrap\_Drop, MenuItem .v..v...

**Sơ đồ lớp fwTouch:**



Hình 4.34 Sơ đồ lớp fwTouch

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| Vector<Touch\*> | m\_Touches | Những điểm cảm ứng chạm vừa chạm. Có giá trị khi loại cảm ứng chạm là cảm ứng đa điểm (Touch All At Once) |
| Touch\* | m\_Touch | Điểm cảm ứng chạm vừa chạm. Có giá trị khi loại cảm ứng chạm là cảm ứng đơn điểm (Touch One By One) |
| bool | m\_SwallowTouches | Kiểm tra cho phép đối tượng cản không truyền tín hiệu xuống nút con |
| Type | m\_TouchType | Loại cảm ứng chạm đc xử dụng |

Bảng 4.47 Thuộc tính lớp fwTouch

**Phương thức:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | onTouchBegan | Touch\* - Tham số 1, Event – Tham số 2 | Tham số 1 trả về dữ liệu của cảm ứng chạm khi bắt đầu chạm, Event trả về thông số điều kiển sự kiện |
| Void | onTouchMoved | Touch\* - Tham số 1, Event – Tham số 2 | Tham số 1 trả về dữ liệu của cảm ứng chạm khi vừa di chuyển điểm chạm, Event trả về thông số điều kiển sự kiện |
| Void | onTouchEnded | Touch\* - Tham số 1, Event – Tham số 2 | Tham số 1 trả về dữ liệu của cảm ứng chạm khi kết thúc chạm, Event trả về thông số điều kiển sự kiện |
| Void | onTouchCancelled | Touch\* - Tham số 1, Event – Tham số 2 | Tham số 1 trả về dữ liệu của cảm ứng chạm khi bị ngắt bởi một lệnh hệ thống có độ ưu tiên cao, Event trả về thông số điều kiển sự kiện |
| Void | onTouchesBegan | Vector<Touch\*> - Tham số 1, Event – Tham số 2 | Tham số 1 trả về dữ liệu của cảm ứng chạm khi bắt đầu chạm lên nhiều đối tượng, Event trả về thông số điều kiển sự kiện |
| Void | onTouchesMoved | Vector<Touch\*> - Tham số 1, Event – Tham số 2 | Tham số 1 trả về dữ liệu của cảm ứng chạm khi di chuyển chạm cho nhiều đối tượng, Event trả về thông số điều kiển sự kiện |
| Void | onTouchesEnded | Vector<Touch\*> - Tham số 1, Event – Tham số 2 | Tham số 1 trả về dữ liệu của cảm ứng chạm khi kết thúc, Event trả về thông số điều kiển sự kiện |
| Void | onTouchesCancelled | Vector<Touch\*> - Tham số 1, Event – Tham số 2 | Tham số 1 trả về dữ liệu của cảm ứng chạm bị hủy bỏ bởi một lệnh có độ ưu tiên cao hơn của hệ thống, Event trả về thông số điều kiển sự kiện |

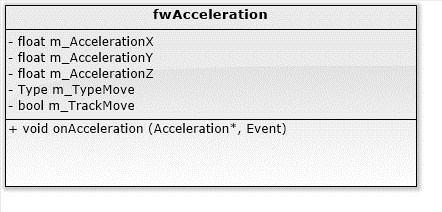
Bảng 4.48 Phương thức lớp fwTouch

4.4.4.4 Cảm ứng gia tốc

Đối với điện thoại di động, xử lý tín hiệu đầu vào duy nhất có thể thay thế được cho cảm ứng chạm là cảm ứng gia tốc. Dù chưa được sử dụng nhiều trong các game nhưng cảm ứng gia tốc đóng góp một phần không nhỏ trong việc xây dựng cảm giác mới cho người chơi.

Cảm biến gia tốc của Cocos2d-x đã hỗ trợ tốt việc quan sát cảm biến. Tuy nhiên, framework eDesign tái sử dựng lại một phần vì các nguyên nhân đã nêu như ở cảm ứng chạm, một phần do muốn hỗ trợ thêm các cơ chế nhận diện di chuyển của điện thoại di động thông qua cảm biến gia tốc.

**Sơ đồ lớp fwAcceleration:**



Hình 4.35 Sơ đồ lớp fwAcceleration

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| float | m\_AccelerationX | Tọa độ trục hoành của gia tốc cảm biến (Acceleration) |
| float | m\_AccelerationY | Tọa độ trục tung của gia tốc cảm biến (Acceleration) |
| float | m\_AccelerationZ | Tọa độ trục cao của gia tốc cảm biến (Acceleration) |
| Type | m\_TypeMove | Loại di chuyển của điện thoại di động |
| bool | M\_TrackMove | Kiểm tra có cần theo dõi thao tác di chuyển của điện thoại di động |

Bảng 4.49 Thuộc tính lớp fwAcceleration

**Phương thức:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | onAcceleration | Acceleration\* - Tham số 1, Event – Tham số 2 | Tham số 1 trả về giá trị của gia tốc cảm biến, tham số 2 truyền về thông số điều khiển sự kiện |

Bảng 4.50 Phương thức lớp fwAcceleration

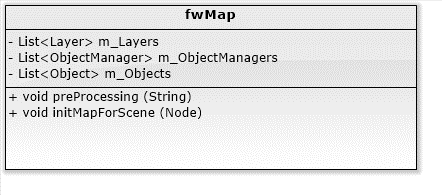
4.4.5 Tính năng bản đồ

“Bản đồ phân ô (TileMap) rất dễ để sử dụng, đủ mềm dẻo để làm việc với Cocos2d-x, dù game của bạn có là một game nhập vai (RPG), vượt địa hình (platformer) hay là thể loại bắn phá (Breakout). Cocos2d-x hỗ trợ bản đồ tạo với phần mềm mã nguồn mở [Tiled Map Editor](http://www.mapeditor.org) và được lưu với định dạng TMX”, được trích dẫn từ trang <http://www.cocos2d-x.org/wiki/TileMap> (ngày 04 tháng 01 năm 2015).

Qua đó có thể thấy, Bản đồ là một phần quan trọng không thể thiếu trong nhiều loại game. Nhưng hiện tại, Cocos2d-x hỗ trợ cơ chế bản đồ chỉ dành riêng cho định dạng TMX. Các định dạng bản đồ khác như .txt, .xml, .csv rất khó để có thể tạo bản đồ trong Cocos2d-x. Thêm vào đó, Định dạng .txt rất quen thuộc với những người mới bắt đầu làm quen với lĩnh vực game do dễ chỉnh sửa, quan sát và có thể thao tác đọc ghi đơn giản.

Framework eDesign thiết kế hỗ trợ với bản đồ định dạng .txt và .xml với cả cơ chế đọc và ghi. EasyDesign trở nên linh hoạt hơn khi thao tác với dạng bản đồ này, và có khả năng, cơ chế tạo bản đồ tích hợp vào trong công cụ hỗ trợ phát triển game EasyDesign.

**Sơ đồ lớp fwMap:**



Hình 4.36 Sơ đồ lớp fwMap

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| List<Layer> | m\_Layers | Tiền xử lý các lớp (layer) trước khi thật sự thêm vào cảnh |
| List<ObjectManager> | m\_ObjectManagers | Tiền xử lý trước các cơ chế quản lý đối tượng trước khi thêm nó vào từng lớp |
| List<Object> | m\_Objects | Tiền xử lý trước các đối tượng trước khi tạo nó thành một nút và thêm vào trong các cơ chế quản lý đối tượng |

Bảng 4.51 Thuộc tính lớp fwMap

**Phương thức:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | preProcessing | String | Truyền vào tên của tập tin chứa các dữ liệu của bản đồ để bản đồ có thể tạo trước các đối tượng |
| Void | initMapForScene | Node | Truyền vào cảnh (scene) cần tạo bản đồ, và tạo các lớp cho cảnh đó. |

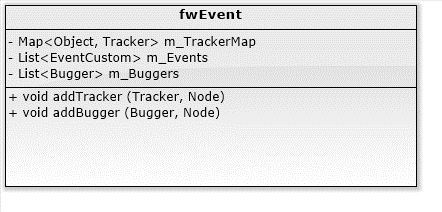
Bảng 4.52 Phương thức lớp fwMap

4.4.6 Tính năng Sự kiện xử lý

Tính năng sự kiện xử lý, fwEvent, được tạo ra nhằm giúp cho người sử dụng công cụ hỗ trợ phát triển game có thể trực quan hơn trong quá trình xây dựng và thiết kế. Bên cạnh đó, Tính năng sự kiện xử lý tạo nên 2 thao tác cơ bản cho một sự kiện là đối tượng quan sát (Tracker) và đối tượng cài đặt (Bugger). Tracker, được cài đặt ở đối tượng nhất định, giúp cho lớp sự kiện xử lý biết được khi nào sự kiện diễn ra. Sau đó, Lớp sự kiện xử lý sẽ kích hoạt các Bugger đã được cài sẵn vào cái đối tượng bị thay đổi bởi sự kiện. Thông qua Bugger các đối tượng đó sẽ thay đổi thuộc tính đúng với quá trình diễn ra của sự kiện.

Về cơ bản, Tracker và Bugger là một đối tượng nút (Node) được thiết kế để bắt và chỉnh sửa dữ liệu từ nút cha. Tuy nhiên, quá trình Tracker quan sát sự kiện ảnh hưởng nhiều đến hiệu suất của game, vì vậy, EasyDesign chưa ứng dụng hẳn tính năng sự kiện xử lý fwEvent.

**Sơ đồ lớp fwEvent:**



Hình 4.37 Sơ đồ lớp fwEvent

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| Map<Object, Tracker> | m\_TrackerMap | Quản lý các đối tượng theo dõi của sự kiện xử lý |
| List<EventCustom> | m\_Events | Quản lý các sự kiện |
| List<Bugger> | m\_Buggers | Quản lý các đối tượng cài đặt |

Bảng 4.53 Thuộc tính lớp fwEvent

**Phương thức:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | addTracker | Tracker, Node | Thêm các đối tượng theo dõi vào các nút đối tượng |
| Void | addBugger | Bugger, Node | Thêm các đối tượng chỉnh sửa vào các nút đối tượng |

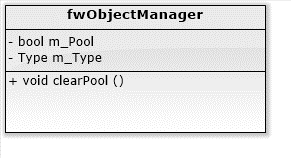
Bảng 4.54 Phương thức lớp fwEvent

4.4.7 Tính năng Quản lý đối tượng

Các danh sách (List) và vector đã được hỗ trợ bởi Cocos2d-x và bản thân thư viện mở của C++. Nhưng, Cấu trúc này đôi khi lại nổi lên vấn đề vốn có của nó: trở nên chậm chạm khi quản lý nhiều đối tượng, ít hỗ trợ cho những game có bản đồ lớn.

Do đó, với sẵn cơ chế nút (Node) có sẵn của Cocos2d-x, framework tinh chỉnh lại tạo thành cấu trúc Tree với mục tiêu chính là hỗ trợ quản lý đối tượng theo 2 loại cây cơ bản: cây nhị phân và cây tứ phân. Ngoài việc đó, framework còn hỗ trợ thêm tính năng hồ chứa (Pool) cho các danh sách và vector.

**Sơ đồ lớp fwObjectManager:**



Hình 4.38 Sơ đồ lớp fwObjectManager

**Thuộc tính:**

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Tên** | **Công dụng** |
| bool | m\_Pool | Kiểm tra cơ chế quản lý có cần ứng dụng hồ chứa (pool) |
| Type | m\_Type | Loại danh sách quản lý đối tượng sẽ sử dụng |

Bảng 4.55 Thuộc tính lớp fwObjectManager

**Phương thức:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu** | **Tên** | **Tham số truyền vào** | **Công dụng** |
| Void | clearPool | Không có | Xóa các đối tượng trong hồ chứa (Pool) |

Bảng 4.56 Phương thức lớp fwObjectManager

Phương thức lấy đối tượng trong danh sách quản lý sẽ được từng đối tượng con của lớp Quản lý danh sách đối tượng thực hiện để phù hợp nhất với từng loại danh sách

CHƯƠNG 5: KẾT QUẢ THỬ NGHIỆM VÀ ĐÁNH GIÁ

5.1 QUY TRÌNH THỬ NGHIỆM.

5.1.1 Các thành phần của hệ thống thử nghiệm.

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

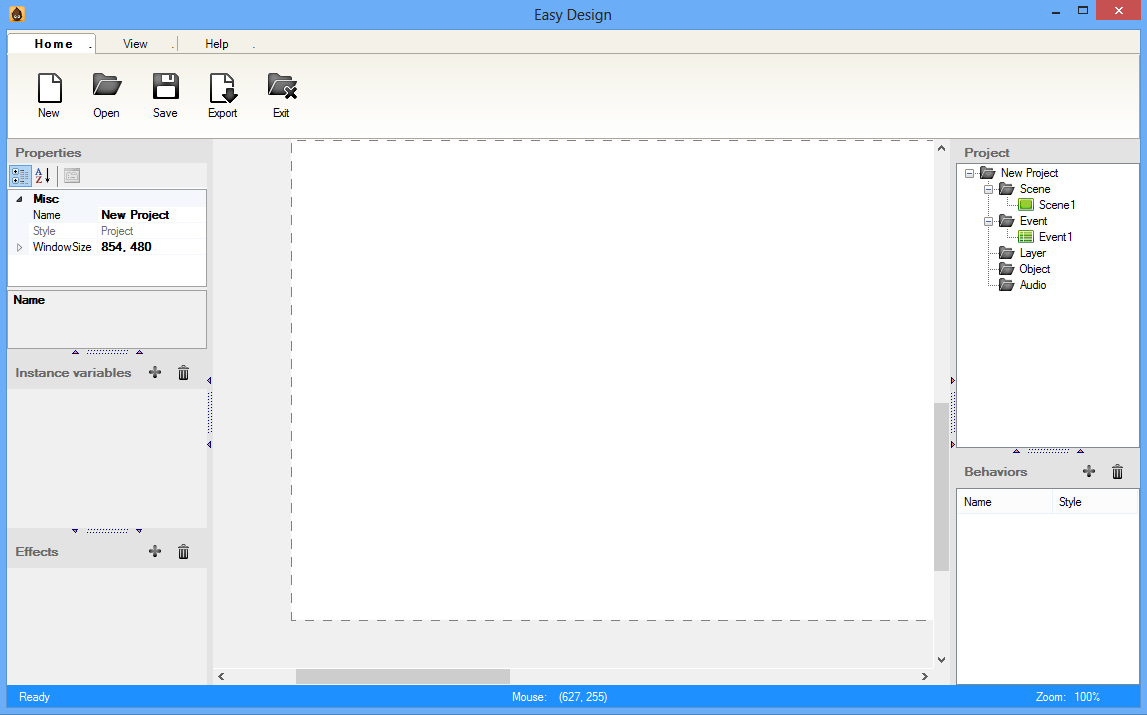
* Chương trình Easy Design
* Hình ảnh
* Visual studio 2012

5.1.2 Tiến hành thử nghiệm.

* Chạy chương trình Easy Design và tạo New Project
* Thêm các đối tượng Sprite và Button
* Xuất file mã nguồn
* Tạo Project Cocos2d-x
* Copy file mã nguồn vào Project Cocos2d-x
* Chạy chương trình bằng Visual Studio.

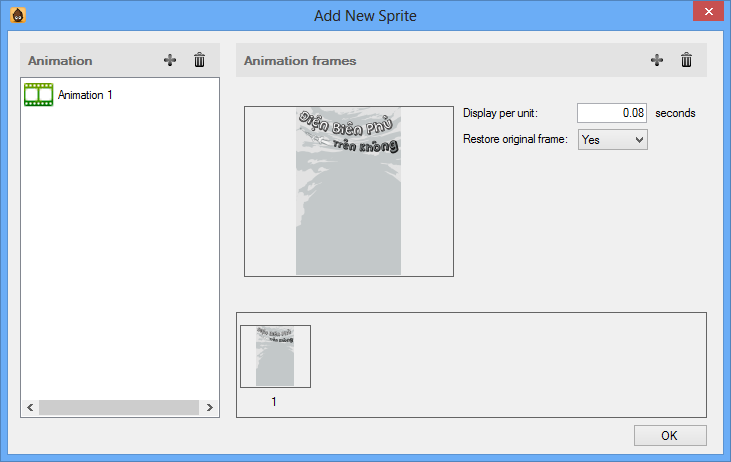
5.2 MỘT VÀI HÌNH ẢNH **KẾT QUẢ THỬ NGHIỆM.**

* Màn hình khởi tạo project mới



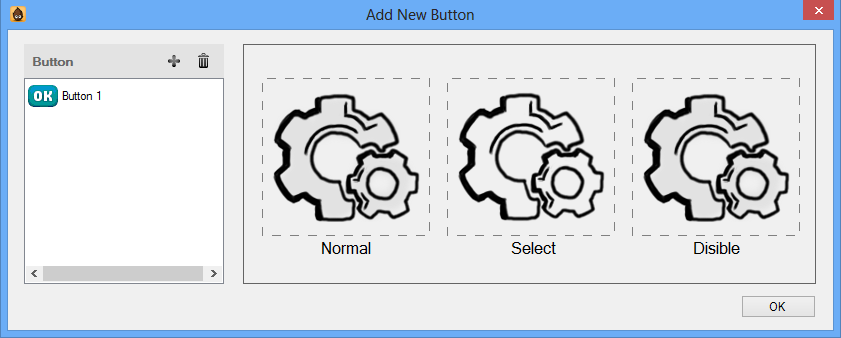
Hình 5.1 Màn hình tạo project mới

Màn hình thêm Sprite



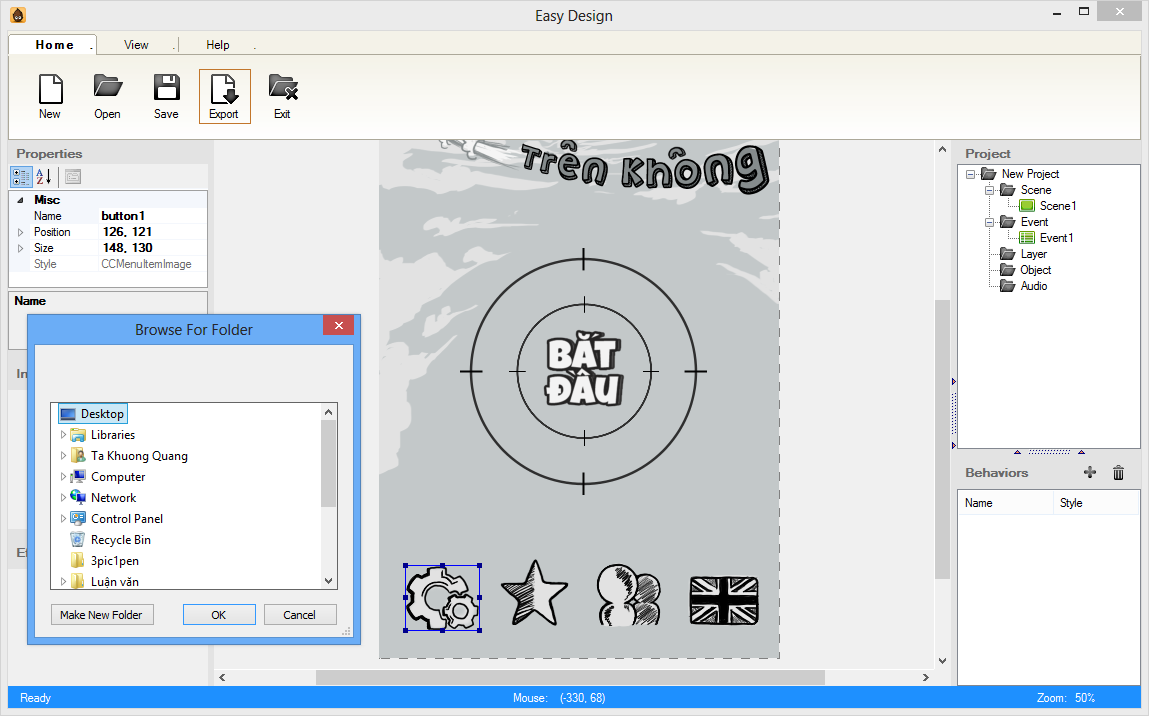
Hình 5.2 Màn hình thêm Sprite

* Màn hình thêm Button



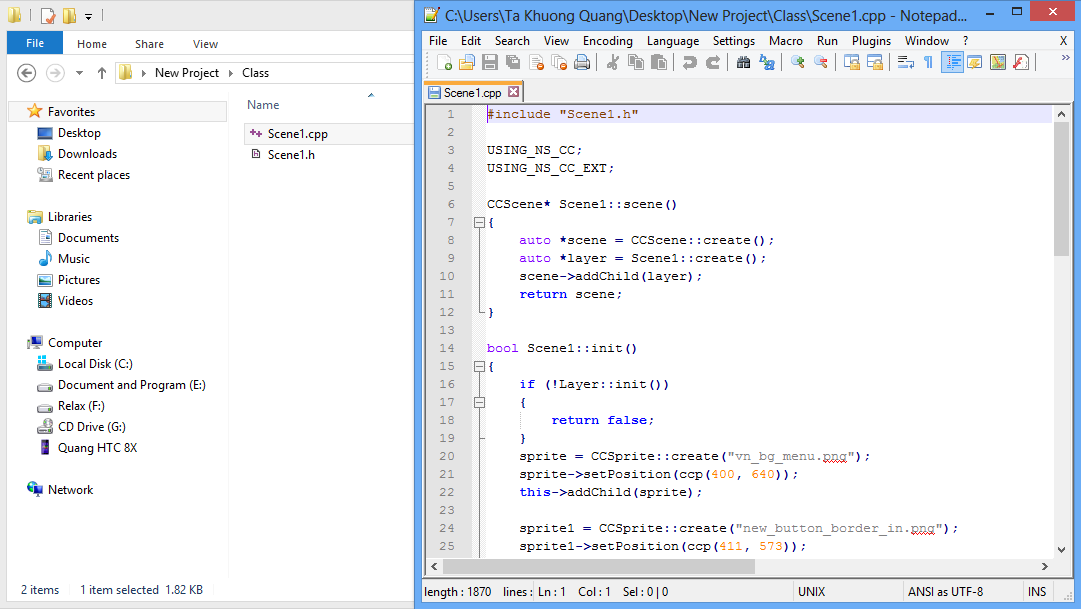
Hình 5.3 Màn hình thêm Button

Màn hình xuất file mã nguồn



Hình 5.4 Màn hình xuất file mã nguồn

* Màn hình file mã nguồn

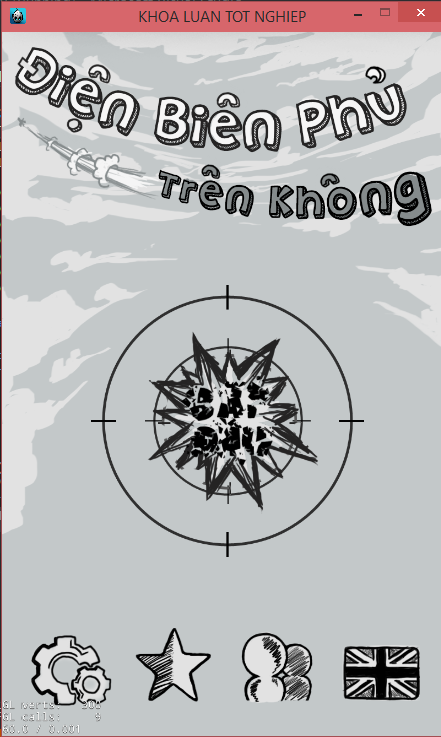


Hình 5.5 Màn hình file mã nguồn

* Màn hình chạy bằng Cocos2d-x



Hình 5.6 Màn hình chạy bằng Cocos2d-x (1)



Hình 5.6 Màn hình chạy bằng Cocos2d-x (2)

CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 KẾT QUẢ ĐẠT ĐƯỢC.

Sau khi tìm hiểu và thực hiện đề tài, Đề tài đã đạt được những kết quả như sau:

* Hiểu được quy trình phát triển game.
* Hiểu được cấu trúc của engine Cocos2d-x.
* Biết cách xây dựng game từ engine Cocos2d-x.
* Biết cách xây dựng framework từ engine Cocos2d-x.
* Biết cách xây dựng chương trình hỗ trợ cho việc phát triển game.

6.2 ƯU ĐIỂM VÀ NHƯỢC ĐIỂM

6.2.1 Ưu điểm:

* Chương trình gọn nhẹ, xử lí nhanh.
* Giao diện trực quan dễ sử dụng.
* Hỗ trợ các đối tượng cơ bản của game.
* FrameWork dễ phát triển.

6.2.2 Nhược điểm:

* Chưa hỗ trợ nhiều loại đối tượng.
* Các hành vi, hiệu ứng còn ít.
* Chưa hỗ trợ việc chỉnh sửa hình ảnh, Chưa hỗ trợ chạy trực tiếp trên IDE.

6.3 HƯỚNG PHÁT TRIỂN

* Hỗ trợ chạy trực tiếp trên IDE mà không phải xuất ra file mã nguồn và chạy bằng Visual Studio.
* Hỗ trợ thêm nhiều hành vi, hiệu ứng cho đối tượng.
* Hỗ trợ thêm nhiều sự kiện giúp việc làm game trở nên dễ dàng hơn.
* Hỗ trợ thêm tính năng tích hợp mạng xã hội Facebook.
* Xây dựng tính năng chỉnh sửa hình ảnh ngay trong IDE.
* Thêm nhiều loại đối tượng trong game
* IDE chạy được trên đa nền tảng, có thể thiết kế ngay trên smartphone.

TÀI LIỆU THAM KHẢO

Tiếng Việt:

[1] Nguyễn Hồng Thuận. “*Nguyên cứu phát triển framework cho game 2D”*. Báo cáo Khóa luận Trường Đại học Công nghệ Thông tin, 2010.

Tiếng Anh:

[2] Roger Engelbert. “Cocos2d-x by Example Beginner's Guide”. Published by Packt Publishing Ltd, April 2013

[3] Sanjay Madhav. “Game Programming Algorithms and Techniques”. Published by Addison – Wesley, 2013

[4] Daniel Sánchez. “Core Techniques and Algorithms in Game Programming”. Published by Crespo Dalmau, September 08, 2003.

[5] Gustavo E Rangel. “Cocos2d-X 3.0: Game Development Programming for iOS and Android”. Published by Dragon Back Game, 2014.

[6] Majed Marji. “Learn to Program with Scratch: A Visual Introduction to Programming with Games, Art, Science, and Math”. Published by No Starch Press, April 2014.

[7] Frahaan Hussain, Arutosh Gurung, Gareth Jones. “Cocos2d-x Game Development Essentials”. Published by Packt Publishing Ltd, August 2014

[8] Siddharth Shekar. “Learning Cocos2d-x Game Development”. Published by Packt Publishing Ltd, September 2014

[9] Randall Hyde. “Write Great Code”, Volume 1. Published by No Starch Press, Octorber 2014.

[10] George T. Heineman, Gary Pollice, Stanley Selkow. “Algorithms in a Nutshell”, Chapter 5, 7. Published by O'Reilly Media, October 2008.

Website tham khảo:

[1] <https://www.scirra.com/manual/1/construct-2>

[2] <http://docs.yoyogames.com/>

[3] <https://help.gamesalad.com/hc/en-us/categories/200134158>

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

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

[7] <http://cocos2d-x.vn/>

[8] <http://developer.kde.org/>

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

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

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

[12] <http://opensource.org/>

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

[14] <http://sourceforge.net/>

Phần mềm tham khảo:

[1] Construct classic, Scirra.

[2] Construct 2, Scirra.

[3] GameMaker, Yoyo games

[4] GameSalad, GameSalad

[5] Scratch, the Lifelong Kindergarten Group