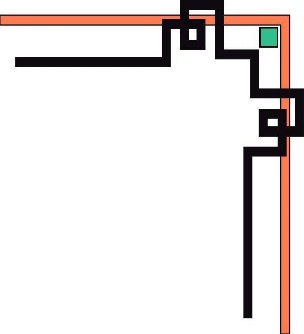
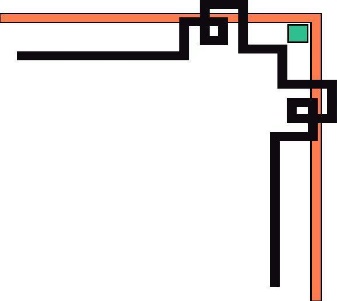
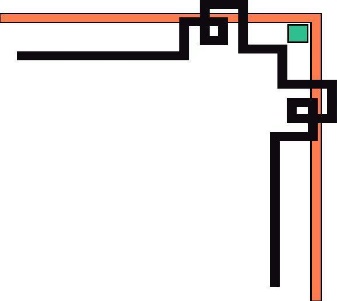
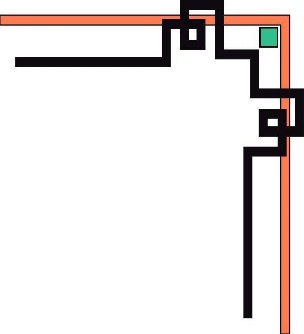
**ĐẠI HỌC QUỐC GIA TP HCM**



**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**

****

**BÁO CÁO**

**CHUYÊN ĐỀ MOBILE AND PERVASIVE COMPUTING**

***ĐỀ TÀI***:

PHÁT TRIỂN GAME MOBILE BẰNG

FRAMEWORK V-PLAY

**Giảng Viên : Nguyễn Trác Thức**

**Sinh Viên Thực Hiện MSSV**

Nguyễn Thị Hương 12520169

Trương Thị Diễm Hương 12520170

***TP. Hồ Chí Minh, Ngày 31 tháng 12 năm 2017***



V-Play game engine

Chuyên đề Mobile and Pervasive Computing

**LỜI NÓI ĐẦU**

Các nhà phát triển game chỉ có thể tạo ra trò chơi của họ trên một nền tảng tại một thời điểm mà thôi. Đó là trước năm 2007 (hoặc trước khi thế hệ iPhone đầu tiên ra đời), khi chỉ có một vài người bị thuyết phục rằng các thiết bị di động có thể cách mạng hóa toàn bộ ngành công nghiệp video game.

Nhưng với tác động mạnh mẽ của các thiết bị di động chạy hệ điều hành khác nhau (Android, iOS, Windows Phone, và BlackBerry OS) và sự ra đời của thế hệ console tiếp theo (như Xbox One, PS4 và Wii U), sự cần thiết để tạo ra một "one-size-fits-all" đã trở thành một tiêu chuẩn trong số các gamedev. Cách duy nhất để thực hiện mục tiêu này là sử dụng cross-platform game engines khi coding một ứng dụng.

Đối với việc phát triển một sản phẩm phần mềm (game, web, ứng dụng,..) nhà phát triển thường quan tâm tới 2 yếu tố lớn: hiệu suất và nền tảng phát triển. Tuy nhiên, trên thực tế, đôi khi đó là một bài toán khó nghĩ đối với người sản xuất phần mềm bởi lẽ:

* + Hiệu suất cao: Dựa trên lĩnh vực ngôn ngữ, những phần mềm phát triển bằng dạng ngôn ngữ biên dịch (C++) thì nhanh hơn so với những ngôn ngữ thông dịch. Tuy nhiên, với những ngôn ngữ biên dịch lại hầu hết phụ thuộc vào thiết bị cài đặt.
  + Nền tảng phát triển: Nói cách khác, đa nền tảng đang là xu hướng chung mà mọi phần mềm hướng tới và nhà phát triển mong muốn sản phẩm của mình làm được.

Làm thế nào để có thể dễ dàng xây dựng một phần mềm vừa có hiệu suất cao, vừa có thể chạy trên nhiều nền tảng khác nhau (iOS, android, …). V-play là một trong những engine tiêu biểu làm được điều đó. Một điểm cộng lớn khác cho engine này đó là: V-play là một engine mã nguồn mở. Nói cách khác, những lập trình viên hoàn toàn có thể sử dụng engine này miễn phí, đồng thời phát triển nó.

Xuất phát từ một 2D Cross – Platform Game – Engine, do công ty V-Play GmbH phát hành version 1.0 vào ngày 12.12.2012 và version 2.0 vào ngày 04.04.2014, thời gian gần đây, bên cạnh game v-play còn hỗ trợ việc phát triển ứng dụng trên mobile trên các nền tảng khác nhau như: iOS, Android, BlackBerry, MeeGo, Sybian, Desktop.

**LỜI CẢM ƠN**

Đầu tiên nhóm xin gửi lời cảm ơn đến thầy Nguyễn Trác Thức. Cảm ơn thầy đã tận tụy truyền dạy kiến thức bổ ích cho nhóm, cho lớp.

Xin cảm ơn các bạn trong lớp đã nhận xét và giúp nhóm test thử game khi release trong những lần release lần đầu.

Tuy cố gắng rất nhiều, nhưng thiếu sót là điều không thể tránh khỏi. Rất mong nhận được sự góp ý của Thầy và các bạn để sản phẩm này có thể hoàn thiện hơn, để sau này nhóm có thể tạo ra những sản phẩm tốt hơn và đưa đến người dùng.

Nhóm xin chân thành cảm ơn!

**NHẬN XÉT CỦA GIẢNG VIÊN**

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Ngày… tháng… năm…..

**Giảng viên**

MỤC LỤC

[**CHƯƠNG I: TỔNG QUAN VỀ MOBILE AND PERVASIVE COMPUTING** 2](#_Toc470886386)

[**1.** **Các khái niệm cơ bản** 2](#_Toc470886387)

[*1.1.* *Mobile Computing* 2](#_Toc470886388)

[*1.2.* *Pervasive Computing* 2](#_Toc470886389)

[*1.3.* *Ứng dụng* 2](#_Toc470886390)

[**CHƯƠNG II: GAME ENGINE** 4](#_Toc470886391)

[**1.** **Game Engine** 4](#_Toc470886392)

[**2.** **Thành phần của Game Engine** 4](#_Toc470886393)

[**3.** **Phân loại Game Engine** 4](#_Toc470886394)

[**4.** **Một số Game Engine thịnh hành** 5](#_Toc470886395)

[**CHƯƠNG III:**  **V-PLAY ENGINE** 7](#_Toc470886396)

[**1.** **Giới thiệu** 7](#_Toc470886397)

[*1.1.* *V-play Game Engine là gì?* 7](#_Toc470886398)

[*1.2.* *Lịch sử hình thành và sự phát triển* 7](#_Toc470886399)

[*1.3.* *Những lợi thế của v-play* 7](#_Toc470886400)

[*1.4.* *Nền tảng hỗ trợ* 7](#_Toc470886401)

[*1.5.* *Cài đặt Qt* 7](#_Toc470886402)

[**2.** **Kiến trúc Component Base Entity** 8](#_Toc470886403)

[*2.1.* *Khái niệm* 8](#_Toc470886404)

[*2.2.* *Ưu điểm* 8](#_Toc470886405)

[**3.** **Các thành phần cơ bản của QT hỗ trợ trong VPlay** 8](#_Toc470886406)

[**CHƯƠNG IV:**  **XÂY DỰNG GAME FLAPPY BIRD VỚI V-PLAY** 10](#_Toc470886407)

[**1.** **Kỹ thuật sử dụng** 10](#_Toc470886408)

[*1.1.* *Render với nhiều chế độ màn hình*. 10](#_Toc470886409)

[*1.2.* *Load resource đa cấu hình* 10](#_Toc470886410)

[*1.3.* *APIs QT* 10](#_Toc470886411)

[*1.4.* *Collision* 10](#_Toc470886412)

[**2.** **Xây dựng kiến trúc game** 11](#_Toc470886413)

[*2.1.* *EntityBase* 11](#_Toc470886414)

[*2.2.* *Scene* 11](#_Toc470886415)

[*2.3.* *Cây thư mục được thiết kế như sau* 11](#_Toc470886416)

[**3.** **Sơ đồ use case** 11](#_Toc470886417)

[**4.** **Thiết kế giao diện** 12](#_Toc470886418)

[*4.1.* *Loading Scene* 13](#_Toc470886419)

[*4.2.* *Menu Scene* 13](#_Toc470886420)

[*4.3.* *GamePlay Scene* 14](#_Toc470886421)

[*4.4.* *GameOver Scene* 14](#_Toc470886422)

[**5.** **Thiết kế độ khó** 15](#_Toc470886423)

[**6.** **Hướng dẫn chơi** 15](#_Toc470886424)

[**CHƯƠNG V:** **CÀI ĐẶT VÀ THỬ NGHIỆM** 16](#_Toc470886425)

[**1.** **Cài đặt** 16](#_Toc470886426)

[**2.** **Triển khai game và phát triển** 16](#_Toc470886427)

[**CHƯƠNG VI:** **TỒNG KẾT** 17](#_Toc470886428)

[**1.** **Đánh giá engine và project** 17](#_Toc470886429)

[*1.1.* *Ưu điểm* 17](#_Toc470886430)

[*1.2.* *Hạn chế* 17](#_Toc470886431)

[**2.** **Hướng phát triển** 17](#_Toc470886432)

[**TÀI LIỆU THAM KHẢO** 18](#_Toc470886433)

**Danh sách các bảng, hình vẽ**

|  |  |
| --- | --- |
| Hình ảnh | Tên |
| Hình 3.1 | Cấu trúc Component Base Entity |
| Hình 4.1: | Cấu trúc thư mục của 1 project Qt |
| Hình 4.2 | Sơ đồ use case |
| Hình 4.3 | Màn hình Loading |
| Hình 4.4 | Màn hình Menu |
| Hình 4.5 | Màn hình play |
| Hình 4.6 | Màn hình gameover |

**CHƯƠNG I: TỔNG QUAN VỀ MOBILE AND PERVASIVE COMPUTING**

1. **Các khái niệm cơ bản**
   1. *Mobile Computing*

Mobile Computing là sự tương tác giữa con người và máy tính, trong đó máy tính có thể sẽ được vận chuyển trong quá trình sử dụng.

* Tính toán di động là “đặt máy tính và tất cả các file cần thiết và phần mềm chung với nhau”. Tính toán di động là bất kỳ loại máy tính nào có liên kết với nhau tạo thành mạng cá nhấn không dây.

Có 3 lớp khác nhau củaMobile Computing:

* Portatble computer: máy tính xách tay: trọng lượng nhẹ bao gồm bàn phím vật tập hợp đầy đủ và chủ yếu được dung như host như: laptop, notebook…
* Mobie phone: Điện thoại di động không chỉ giới hạn giao tiếp bằng giọng nói, điện thoại phổ thông, điện thoại thông minh, phonepad…
* Wearable: máy tính có thể mặc deo được: hầu hết được giới hạn một số chức năng chủ yếu như: đồng hồ, kính, dây đeo tay…
  1. *Pervasive Computing*
* Pervasive computing chính là tương lai của mạng máy tính, cho phép truyền thông không dây giữa các cá nhân và các thiết bị máy tính và giữa các thiết bị chính nó.
* Máy tính phổ biến chính là tương lai của mạng máy tính cho phép truyền thông không dây giữa các cá nhân và các thiết bị máy tính và giữa các thiết bị chính nó.
* Các thiết bị máy tính phổ biến là rất nhỏ, với các vi xử lý ẩn bên trong và được nhúng vào trong bất kỳ loại đối tượng nào như quần áo, nhà, xe hơi, đồng hồ,…
* Còn được gọi là Ubiquitous Computing
* Mục đích của Pervasive Computing là dành cho máy tính có sẵn bất cứ nơi nào cần thiết. Nó trải rộng tính thông minh và kết nối tất cả mọi thứ:
  + Tàu thủy,Máy bay, Ô tô, cầu, đường hầm, máy, tủ lạnh, cánh cửa, thiết bị chiếu sáng, Giày dép, nón, Công cụ, Nhà và thậm chí cả những thứ như cốc cà phê và những con chip gắn trên cơ thể con người
  + Truy cập Internet di động,truyền thông không dây, các thiếtbị cầm tay, và Bluetooth đã làm cho pervasive computing trở thành hiện thực.
  1. *Ứng dụng*
* Hệ thống Định vị Toàn cầu ([tiếng Anh](http://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh): *Global Positioning System* - **GPS**) là hệ thống xác định vị trí dựa trên vị trí của các [vệ tinh nhân tạo](http://vi.wikipedia.org/wiki/V%E1%BB%87_tinh), do [Bộ Quốc phòng Hoa Kỳ](http://vi.wikipedia.org/wiki/B%E1%BB%99_Qu%E1%BB%91c_ph%C3%B2ng_Hoa_K%E1%BB%B3) thiết kế, xây dựng, vận hành và quản lý. Trong cùng một thời điểm, tọa độ của một điểm trên mặt đất sẽ được xác định nếu xác định được khoảng cách từ điểm đó đến ít nhất ba vệ tinh.
* Nó có thể được sử dụng như một thiết bị định vị của xe, bảo vệ người dùng khỏi bị thất lạc. Khi một thiết bị giám sát GPS được nhúng vào điện thoại di động, nó được sử dụng cho việc tìm kiếm hướng, tính toán khoảng cách, và nhiều hơn nữa. Nó cũng có thể được sử dụng để xác định vị trí của một chiếc xe bị đánh cắp hoặc vật nuôi bị mất.
* Những lợi thế khác của hệ thống định vị toàn cầu GPS là tính tương thích Google Earth, dễ dàng sử dụng, nhỏ và dễ dàng che giấu, tính chính xác trọng việc theo dõi, và sức mạnh của nam châm của nó cho phép nó được đặt dưới gầm xe và nó có khả năng chịu nước.
* Một điều khác về GPS là nó giúp cha mẹ theo dõi nơi đến của con em họ. Nó có thể được mang theo bởi một đứa trẻ hay được đặt trong một chiếc xe của teen để thông báo cho cha mẹ vị trí chính xác của đứa trẻ. Điều này rất hữu ích khi xảy ra trường hợp khẩn cấp. Hệ thống tự động gửi thông tin cho phụ huynh khi trường hợp như vậy xảy ra.
* Về kinh doanh. GPS cũng có lợi cho các doanh nghiệp đặc biệt là các công ty lớn. Thiết bị có thể dễ dàng theo dõi bất kỳ loại phiền toái hay phiền nhiễu trong địa điểm kinh doanh của họ. Điều này giúp các quan chức dễ dàng quản lý nguồn nhân lực cũng như nguồn tài nguyên.

**CHƯƠNG II: GAME ENGINE**

1. **Game Engine**

Game Engine là một phầm mềm được thiết kế đặc biệt để xây dựng và phát triển video game. Game Engine chính là thành phần cốt lõi làm nền móng để xây dựng nên Game. Sự khác biệt giữa Game Engine và bản thân Game tương tự như như sự khác biệt giữa động cơ và chiếc xe. Chúng ta có thể đem động cơ ra khỏi xe, chỉnh sửa và dùng lại. Một Game Engine có thề được coi là một phần không phụ thuộc vào Game, nó cũng có thể được thay đổi độc lập với Game mà không làm thay đổi cấu trúc logic của Game.

1. **Thành phần của Game Engine**

Các thành phần cơ bản của 1 Game Engine bao gồm

* Hệ thống toán học (Math system)
* Hệ thống xử lý tập tin dữ liệu (File system)
* Hệ thống hiển thị đồ hoạ (Render system)
* Hệ thống quản lý hoạt họa (Animation system)
* Hệ thống xử lý nhập và xuất (Input and Output system)
* Hệ thống xử lý các tương tác vật lý (Physics system)
* Hệ thống xử lý trí tuệ nhân tạo (Artificial intelligence system)
* Hệ thống xử lý mạng (Network system)
* Hệ thống tạo hiệu ứng khác như khói lửa, âm thanh, …(Effect system)

1. **Phân loại Game Engine**

Có 3 tiêu chí để phân loại Game Engine:

* Theo Level:
  + High – level
  + Low – level
  + ….
* Theo tính năng
  + Isometric Engine: Dùng trong các trò chơi 2D, thường được sử dụng trong các game chiến lược và trong các thể loại game nhập vai: Baldur’s Gate, Diablo 2, Warlords Battle Cry 2
  + 3D FPS (First Person Shoot) Engine: dùng phổ biến trong các game 3D hiện nay; hỗ trợ nhiều chức năng như tô bóng điểm (pixel shader), quản lý diễn hoạt (animation), mô phỏng vật lý, mô phỏng chuyển động của các hạt nhỏ (như bụi, lửa, khói), mô phỏng chất lỏng, áo quần, và các hiệu ứng khác. Một số game viết trên engine này là: Doom, Quake, Half life,…
  + MMOG (Massive Multiplayer Online Game) Engine : dựa trên việc lập trình mạng và quản lý dữ liệu thông qua mạng, xử lý cho một số lượng lớn người chơi trong cùng 1 lúc. Một số game viết trên engine này như: Ever Quest, Asheron’s Call, Anarchy Online.
* Theo license
  + Open source
  + Commercial
  + …

1. **Một số Game Engine thịnh hành**

|  |  |
| --- | --- |
| Game Engine | Mô tả |
| Unity | Unity cho phép bạn tạo ra các trò chơi của bạn bằng cách thao tác các đối tượng và các yếu tố trong 3D và gắn nhiều thành phần vào chúng (như hiệu ứng hình ảnh, bóng, vv…). Phát triển một ứng dụng bằng cách sử dụng engine yêu cầu tất cả các scripts được viết bằng C #, UnityScript, hoặc Boo và gắn vào đối tượng 3D như là các thành phần.  Phiên bản miễn phí cũng hỗ trợ đầy đủ cho việt phát triển Windows, Mac OS X, Linux, và các nền tảng di động lớn (iOS, Android). Phiên bản pro (mà đòi hỏi một license) thì hỗ trợ thêm cho PlayStation, Xbox 360 và Nintendo Wii.  Một số game: Bad Piggies, Temple Run Oz,… |
| V-play | Tập trung vào việc phát triển nhanh chóng các game 2D. Nó cung cấp cho bạn với một tài liệu được cập nhật thường xuyên, code mẫu, và lời khuyên hữu ích để giúp bạn. Trong thực tế, các khoảng thời gian trung bình mà bạn có thể tạo ra một trò chơi cho điện thoại di động, máy tính bảng, hoặc máy tính bàn với hiệu suất thực sẽ chỉ mất khoảng 3 ngày.  Một số game: Trix, Chicken Outbreak,… |
| Marmalade | Engine cho phép bạn phát triển game và các ứng dụng trong C/C++ và deloy cho cả hai điện thoại di động và máy tính bàn. Hỗ trợ iOS, Android và BlackBerry.  Marmalade cho phép bạn dễ dàng chuyển đổi các project của bạn cho cả Android, iOS, Windows Phone, Tizen, BlackBerry, Windows OS, và Mac OS chỉ từ một source code duy nhất. Nó cũng cũng cho phép bạn tích hợp các thư viện chuẩn và thư viện của bên thứ ba mà không cần virtual machines (như Java VM) hoặc cross-language translation.  Một số game: Draw Something, Zombies,… |
| Corona SDK | Một platform cũng khá phổ biến, cho phép bạn tạo ra các game (cũng như các ứng dụng khác) cho iOS và Android và WP. Họ tuyên bố là đã được hơn 150.000 developer sử dụng.Emulator rất nhanh và refresh gần như tức thì mỗi khi bạn chỉnh sửa một tập tin trong project. Ngoài ra, lập trình viên có thể dễ dàng điều chỉnh các độ phân giải màn hình. Tuy nhiên, về mặt plugin và game cần có tính năng kết nối mạng và chi phí cho engine này khoảng 600$/năm.  Một số game: Fun Run, Blast Monkey, Freeze. |
| Cocos2D JS | Cocos 2d JS là một game engine mã nguồn mở được thiết kế để làm game 2D/2.5D. Game code được viết hoàn toàn nằng JavaScript. Một số nền tảng: iOS, Android, Blackberry, Windows Phone, Mac, Windows và Linux.  Một số game: Contra: Evolution (Konami), MT Online, ... |

**CHƯƠNG III: V-PLAY ENGINE**

1. **Giới thiệu**
   1. *V-play Game Engine là gì?*

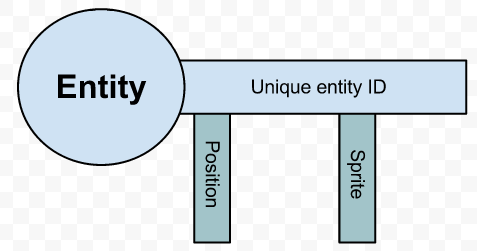
* V-play Game Engine là một engine được phát triển dựa trên Qt Framework, dùng để phát triển game hoặc ứng dụng trên cả mobile và desktop.
* Mục đích của v-play là giúp lập trình viên có thể dễ dàng tạo ra những phần mềm chạy được trên nhiều nền tảng khác nhau từ một mã nguồn.
  1. *Lịch sử hình thành và sự phát triển*
* Ngày 12/12/2012, v-play Engine phiên bản 1.1 được ra mắt. Phiên bản chạy ổn định hiện nay là 2.0.3 (30/6/2014).
* Từ những ngày đầu ra mắt, v-play được biết tới như một game engine. Tuy nhiên, thời gian gần đây, engine này hỗ trợ cả việc phát triển các ứng dụng trên mobile.
  1. *Những lợi thế của v-play*
* V-play Engine cung cấp tập hợp lớn các component bao gồm: UI, mutimedia, networking. Và tất cả những component này có sẵn trong QML (ngôn ngữ engine này sử dụng) hoặc được thực thi bời C++. Việc hiện thực bằng C++ đảm bảo yếu tố hiệu suất của sản phẩm phần mềm làm ra
* Việc sử dụng Qml đảm bảo về sự tiện lợi không chỉ về mặt giao diện mà còn về công sức và chi phí thời gian để phát triển ứng dụng của lập trình viên.
  1. *Nền tảng hỗ trợ*

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Mobile | Android | Từ Android 2.3 trở lên |
| iOS | Apple iOS version 5.0 trở lên  Inphone 3GS, Ipad, iPod touch 3rd Gen hoặc mới hơn |
| Blackberry 10 |  |
| Symbian |  |
| MeeGo |  |
| Desktop | Windows | 32 bit hoặc 64 bit  Window XP Service Pack 2 trở lên |
| Mac OS X | 64 bit Apple Mac OS X 10.6 trở lên |
| Linux | 32 bit hoặc 64 bit Linux Ubuntu 12.04, 12.10, 13.10 hoặc OpenSuse 12 |

* 1. *Cài đặt Qt*
* Download và cài đặt theo hướng dẫn tại link: https://v-play.net/download/

1. **Kiến trúc Component Base Entity**
   1. *Khái niệm*

***Play hướng developer lập trình theo kiến trúc component-base entity.***

Kiến trúc Component là hướng lập trình hướng dữ liệu (Document-Oriented). Trong đó một đối tượng thường được gọi là entity (thực thể). Các thực thể là những đối tượng rỗng, không chứa bất kỳ dữ liệu nào riêng biệt và không có bất kỳ sự khác nhau nào giữa chúng. Một thực thể có thể gắn thêm nhiều component khác nhau để thực hiện các mục đích riêng. Mỗi thực thể sẽ mang những thuộc tính và hành vi của component được thêm vào, khi đó các thực thể mới thực sự khác nhau.

**Hình 3.1: Cấu trúc Component Base Entity**

Trong thực thể ở trên, có các component là: Position, Sprite.

* Mỗi component thông thường sẽ có hai thành phần là:
  + Attributes: dữ liệu riêng của component.
  + Behaviours: các hàm xử lý các dữ liệu.
* Mỗi component trong một thực thể sẽ hoàn toàn độc lập với nhau và được thực thi trên những hệ thống riêng biệt, chính vì thế, tốc độ thực thi của kiến trúc Component nhanh hơn kiến trúc cây thừa kế rất nhiều.
  1. *Ưu điểm*
* Giảm độ phức tạp của dự án.
* Tốc độ xử lý cực nhanh.
* Dễ vận hành và bảo trì game
* Dễ dàng cho các dự án có nhiều người cùng tham gia khi mỗi người có thể quản lý mỗi component riêng biệt.

1. **Các thành phần cơ bản của QT hỗ trợ trong VPlay**

|  |  |
| --- | --- |
| Module | Description |
| Qt core | Chứa các thành phần cơ sở, bao gồm các containers (tương tự thư viện chuẩn của các ngôn ngữ lập trình: Qvector, Qlist, Qmap...), các thư viện nhập xuất đa nền tảng, lập trình đa luồng (threading) và xử lý song song (concurrency) |
| Qt GUI | Thành phần chính để lập trình giao diện |
| Qt Widgets | Các widget. Button, hộp thoại và những thứ tương tự trong giao diện đều gọi là các widget. |
| Qt QML | Hỗ trợ cho QML và Javascript. QML (dựa trên Js) là một ngôn ngữ để viết giao diện cho ứng dụng Qt, bên cạnh ngôn ngữ lập trình chính C++ |
| Qt Quick | QML là một phần của Qt Quick |
| Qt Quick Controls | Các widget điều khiển dành cho Qt Quick (dành cho Desktop) |
| Qt Quick Layouts | Bố trí widget trong Qt Quick |
| Qt Network | Network abstraction layer (NAL). Đầy đủ các giao thức TCP, UDP, HTTP và SSL |
| Qt Multimedia | Thư viện về âm thanh, hình ảnh, camera.. |
| Qt Multimedia Widgets | Các widget dành cho Qt Multimedia. |
| Qt SQL | Module làm việc với cơ sở dữ liệu |
| Qt Webkit | Webkit là một module cho các web browser dùng để hiển thị các trang web. Qt WebKit là một thư viện giúp bạn làm việc với WebKit. |
| Qt WebKit Widgets | Các widget cho Qt WebKit |
| Qt Test | Hỗ trợ Unit Test cho các ứng dụng Qt. |

**CHƯƠNG IV: XÂY DỰNG GAME FLAPPY BIRD VỚI V-PLAY**

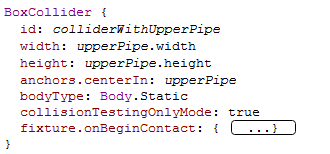
1. **Kỹ thuật sử dụng**
   1. *Render với nhiều chế độ màn hình*.

* Có rất nhiều cơ chế để đảm bảo hình ảnh của bạn có thể render ra toàn màn hình device.
  + Cơ chế scale chỉ resource từ lớn về nhỏ hoặc từ nhỏ đến lớn.
  + Sử dụng tất cả resrouce dạng nhỏ và scale toàn thế giới
  + Sử dụng resource dạng vector.
  + Sử dụng nhiều chế độ resource.
* Giống như các engine cocos2d hay GameMaker, Vplay hỗ trợ cơ chế render cho nhiều kích thước màn hình. Sử dụng cơ chế nhiều dạng resource hoặc dạng scale toàn thế giới. Bạn có thể tùy chọn để phù hợp với dự án của mình.
* Bạn có thể xem chi tiết tại đây: http://v-play.net/doc/vplay-different-screen-sizes/
  1. *Load resource đa cấu hình*
* Bằng cách sử dụng cơ chế đợi, chương trình của bạn sẽ chỉ thực thi khi tất cả các resource cần thiết đã load xong.
* Điều này nghe có vẻ bất hợp lý, và tất nhiên sẽ ảnh hưởng đến performance của hệ thống, tuy nhiên nó đảm bảo rằng resource sẽ được load đầy đủ trong game với các máy có cấu hình thấp.
* VPlay cũng hỗ trợ cơ chế pooling, cho phép resource không bị load lại hay các đôi tượng khởi tạo lại nhiều lần. Để biết rõ hơn, bạn có thể xem thêm mẫu thiết kế **Pooling Object Design Pattern**.
  1. *APIs QT*

VPlay cũng hỗ trợ các UI component mà QT hỗ trợ cho QT Application. Trong đồ án của mình, chúng tôi có sử dụng ImageButton. Đây là một dạng Button có chứa hình ảnh, có đầy đủ các thuộc tính của Button nhưng hỗ trợ thêm Image.

* 1. *Collision*

Vplay sử dụng phương pháp xét va chạm hình vuông. Thông qua component BoxCollider.



BoxCollider có hai kiểu là dạng Static và Dynamic. Static là dạng bounding cho vật đứng yên trong thế giới, Dynamic là dạng bouding dành cho vật chuyển động.

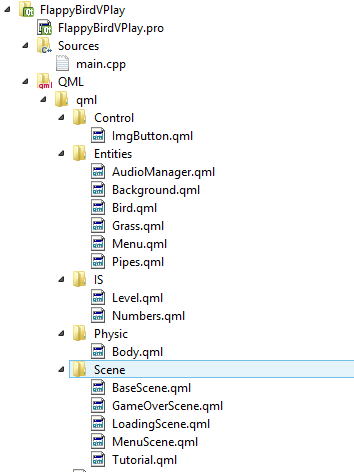
Sau khi va chạm thì hàm onBeginContact sẽ được gọi. Việc của bạn là viết logic va chạm trong hàm này.

1. **Xây dựng kiến trúc game**
   1. *EntityBase*

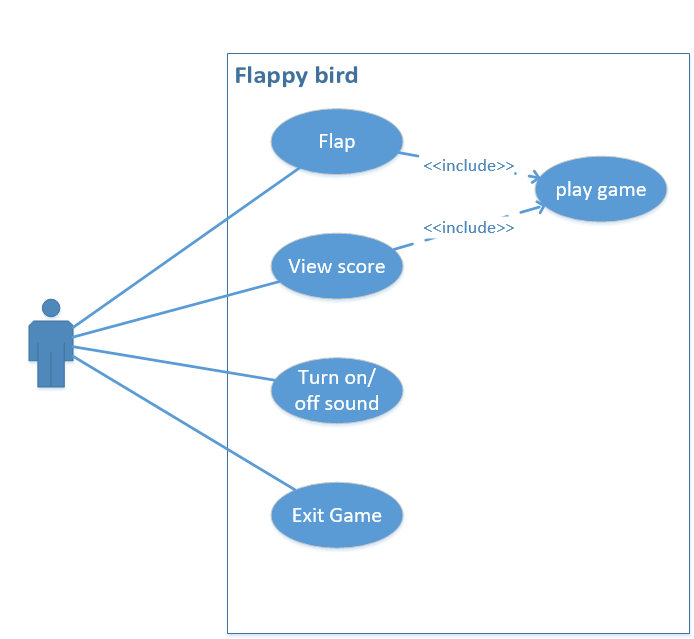
* Các object được tạo ra trong game đều phải kế thừa Entity Base.
* EntityBase là một component của Vplay, trong đó chưa các attribute và các function cơ bản như id, entityID….
* Việc kế thừa từ EntityBase giúp dễ dàng điều khiển và cập nhật thuộc tính trong các đối tượng hơn.
  1. *Scene*

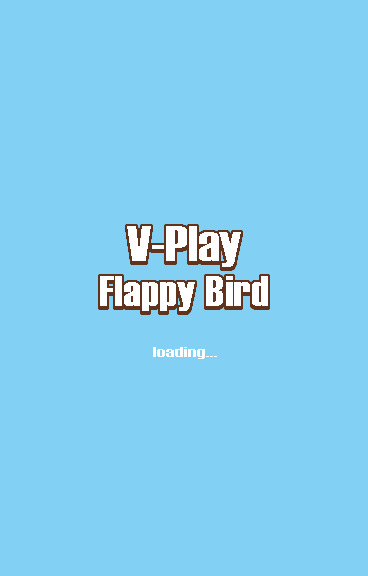
Scene trong game cũng được kế thừa từ EntityBase. Nhưng mà thay vì viết trực tiếp các scene từ EntityBase, ta sẽ viết lại thành một lớp SceneBase. Đây sẽ là component cho tất cả các scene tạo ra sau này.

* 1. *Cây thư mục được thiết kế như sau*

**Hình 4.1: Cấu trúc thư mục của 1 project Qt**

1. **Sơ đồ use case**

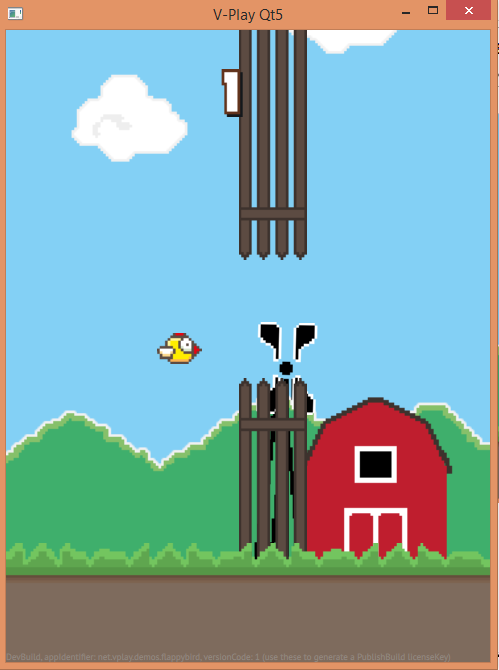
**Hình 4.2:** **Sơ đồ use case**

1. **Thiết kế giao diện**
   1. *Loading Scene*

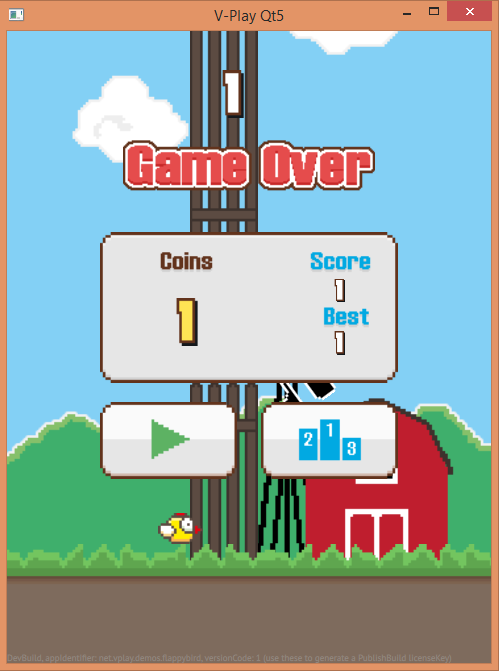
**Hình 4.3: Màn hình Loading**

* 1. *Menu Scene*

**Hình 4.4: Màn hình Menu**

* 1. *GamePlay Scene*

**Hình 4.5: Màn hình play**

* 1. *GameOver Scene*

**Hình 4.6: Màn hình gameover**

1. **Thiết kế độ khó**

* Độ khó được thiết kế dựa vào tần suất xuất hiện của các pipe. Ở cấp độ đầu, các pipe sẽ xuất hiện trong 70 pixel, tuy nhiên càng về sau thì tần suất xuất hiện càng dày đặc hơn.
* Điều này đòi hỏi người chơi sẽ phải khéo léo hơn khi điều khiển chú chim của mình qua được lỗ giữa 2 pipe. Lỗ này gọi là gateway.
* Các gateway sẽ không giảm kích thước theo thời gian mà sẽ thay đổi độ cao. Với các level khó thì độ chênh lệch của 2 gateway gần nhau sẽ tăng lên, lúc đó chú chim sẽ phải di chuyển nhiều hơn để qua được gate tiếp theo.

1. **Hướng dẫn chơi**

* Đây là một tựa game nổi tiếng, điều khiển con chim của bạn đi qua các gateway.
* Gateway là lỗ nhỏ giữa hai rào chắn.
* Khi bạn chạm vào màn hình, chú chim sẽ bay lên hướng chéo với một vận tốc xác định. Khéo léo điều khiển chú chim qua 1 gateway, bạn sẽ được một điểm. Cố gắng đạt điểm cao hơn trong danh sách bạn bè của bạn và tận hưởng niềm vui chiến thắng.
* Việc đánh vào tâm lý chiến thắng sẽ giúp người chơi có được sự nỗ lực cần thiết. Đây là điểm hấp dẫn nhất mà chúng tôi quyết định thực hiện tựa game này.

**CHƯƠNG V: CÀI ĐẶT VÀ THỬ NGHIỆM**

1. **Cài đặt**

Game demo có bản chạy trên desktop và được chạy thử trên device (môi trường android). Một số hình ảnh khi triển khai trên device:

1. **Triển khai game và phát triển**

* Phát triển ứng dụng hay game bằng v-play cần download framework như hướng dẫn phía trên và develop.

**CHƯƠNG VI: TỒNG KẾT**

1. **Đánh giá engine và project**
   1. *Ưu điểm*
   * Engine dễ sử dụng, phù hợp phát triển những project đòi hỏi hoàn thành trong thời gian ngắn.
   * Xử lý được những yếu tố của 1 game cơ bản nhưng: load hình ảnh game, xử lý va chạm, …
   1. *Hạn chế*

* Chưa sử lý được phần âm thanh game
* Giao diện chưa thu hút

1. **Hướng phát triển**

Phát triển game trên Qt cần lưu ý một số điểm như sau:

* Cần hiểu rõ ngôn ngữ Qt, cấu trúc hoạt động
* Ngôn ngữ lập trình C++

**TÀI LIỆU THAM KHẢO**

<https://v-play.net/doc/>  
<https://blog.knoldus.com/2014/10/04/tutorial-1/>