**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HÒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**

**-----oOo-----**



**PHÁT TRIỂN PHẦN MỀM**

**MÃ NGUỒN MỞ**

**ĐỀ TÀI:**

**TÌM HIỂU MÃ NGUỒN MỞ MONOGAME**

**“GAME PLANTS VS ZOMBIES”**

***GVHD: SVTH:***

**GS. TS Vũ Thanh Nguyên 10520272 – Lưu Quang Vũ**

**10520378 – Nguyễn Giang Châu**

**10520464 – Ngô Duy Kha**

*TP. Hồ Chí Minh, ngày 08 tháng 01 năm 2014.*

# **MỞ ĐẦU**

Hiện nay, ngành công nghệ thông tin ngày càng phát triển, đi kèm với nó là việc ứng dụng công nghệ thông tin vào hầu hết các lĩnh vực khác cũng phát triển theo. Các ứng dụng nền tảng di động phát triển nhanh một cách chóng mặt.

Trên thị trường ngày nay, có rất nhiều thiết bị di động ra đời. Việc tìm hiểu và lập trình trên tất cả là một bài toán khó khăn cho một lập trình viên. Vì họ cần phải biết nhiều ngôn ngữ, nền tảng để làm việc được trên tất cả chúng. Nhưng rất may mắn, việc lập trình đa nền tảng cũng phát triển theo, nó giúp người lập trình viên có thể giảm bớt thời gian một cách đang kể khi làm việc chỉ với một ngôn ngữ, một môi trường nhưng lại tạo ra được nhiều sản phẩm chạy trên các nền tảng khác nhau.

Để tìm hiểu vấn đề trên, nhóm quyết định chọn đề tài “Tìm hiểu MonoGame với ứng dụng game Plants Vs Zombies” nhằm mục đích chính là tìm hiểu phương thức hoạt động của việc lập trình đa nền tảng trên thiết bị di động.

Đề tài được thực hiện trong vòng 3 tháng bao gồm tất cả các giai đoạn: phân tích, thiết kế, lập trình và báo cáo. Vì thời gian quá ít, nên việc sai sót là điều không thể tránh khỏi. Đồng thời, ứng dụng chỉ chạy được trên 2 nền tảng chính là Windows Phone và Android, các thiết bị iOS chưa hỗ trợ.

***Nhóm sinh viên thực hiện.***

# **NHẬN XÉT**

**(của GVHD)**

*TP. HCM, ngày tháng năm 2014.*

**GVHD**

**GS. TS. Vũ Thanh Nguyên**

# **MỤC LỤC**

[MỞ ĐẦU ii](#_Toc376918979)

[NHẬN XÉT iii](#_Toc376918980)

[MỤC LỤC iv](#_Toc376918981)

[Chương 1: GIỚI THIỆU MONOGAME 1](#_Toc376918982)

[I. MONOGAME LÀ GÌ? 1](#_Toc376918983)

[II. CÀI ĐẶT MONOGAME? 2](#_Toc376918984)

[III. CẤU TRÚC MONOGAME? 3](#_Toc376918985)

[Chương 2: CÁC GIẤY PHÉP SỬ DỤNG 5](#_Toc376918986)

[I. GIẤY PHÉP Ms-PL 5](#_Toc376918987)

[II. GIẤY PHÉP GPL 6](#_Toc376918988)

[III. GIẤY PHÉP MIT 6](#_Toc376918989)

[IV. GIẤY PHÉP APACHE. 7](#_Toc376918990)

[Chương 3: THIẾT KẾ HỆ THỐNG 8](#_Toc376918991)

[I. MÔ HÌNH USECASE 8](#_Toc376918992)

[II. MÔ HÌNH CLASS 9](#_Toc376918993)

[Chương 4: CÀI ĐẶT MINH HOẠT 15](#_Toc376918994)

# **Chương 1: GIỚI THIỆU MONOGAME**

Ngày nay, nhiều thiết bị di động lần lượt ra đời như Lumia với hệ điều hành Windows Phone, các Tablet chạy hệ điều hành Windows 8 Metro, các thiết bị Samsung, Sony chạy hệ điều hành Android của Google; các thiết bị iPhone, iPod với hệ điều hành iOS.

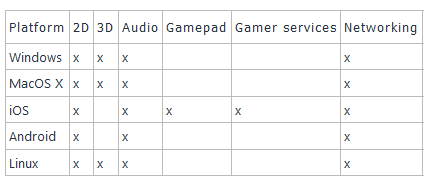
Với nhu cầu sử dụng rộng lớn các thiết bị di động như ngày nay, các lập trình viên gặp rất nhiều khó khăn trong việc chọn lựa nền tảng cũng như ngôn ngữ lập trình.

Để giải quyết những khó khăn đó, MonoGame ra đời. Mục tiêu chính giúp các lập trình viên có thể phát triển ứng dụng từ nền tảng Windows Phone của mình chạy trên các thiết bị với hệ điều hành khác mà không cần phải viết lại game.

1. **MONOGAME LÀ GÌ?**

MonoGame là một dự án mã nguồn mở được phát triển trên nền tảng của Microsoft XNA 4 Framework. Với khẩu hiệu “MonoGame -Write Once, Play Everywhere”, mục tiêu của MonoGame là cho phép các XNA developers có thể mang game của mình sang nhiều nền tảng khác nhau mà không cần phải viết lại game.

Hiện tại, MonoGame đã được hỗ trợ trên một số nền tảng nổi tiếng như: Xbox 360, Windows & Windows Phone, iOS, Android, Mac OS X, Linux và Windows 8 Metro.



Nhìn bảng trên, ta có thể thấy MonoGame ngày càng được hoàn thiện và hỗ trợ hầu như toàn bộ các kỹ thuật khó khăn trong game như: xử lý 3D, Networking...

MonoGame đang dần trở nên phổ biến và trở thành một công cụ mạnh mẽ cho các lập trình viên sử dụng để chuyển đổi nền tảng co các ứng dụng và đặc biệt hoàn toàn miễn phí.

Một số game được viết trên nền tảng MonoGame:

* Đã có mặt trên iOS App Store:



* Đã có mặt trên Android Maket:



Về mã nguồn, MonoGame được phát triển tại codeplex và mã nguồn đầy đủ tại:

* Trang chủ: <http://monogame.codeplex.com/>
* Mã nguồn: <https://code.google.com/p/monoxna/>

1. **CÀI ĐẶT MONOGAME?**
2. **Windows:**

Để phát triển trên Windows, điều kiện tối thiểu gồm:

1. Visual Studio 2010 Express or Above or MonoDevelop/Xamarin Studio
2. XNA 4.0 Game Studio
3. Windows Phone 7 SDK
4. OpenAL installed.

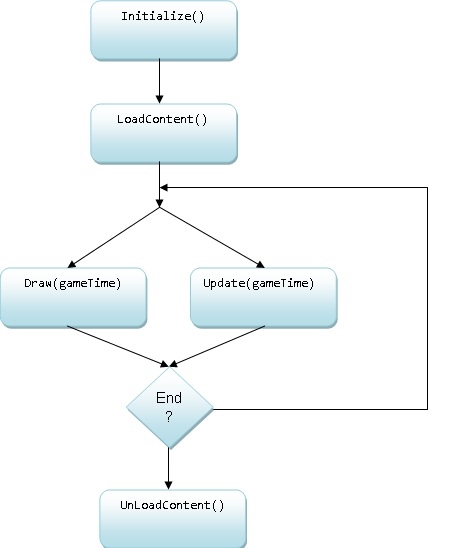
Nếu muốn chuyển đổi chạy trên nền tảng Windows RT hoặc Windows Phone 8:

1. Visual Studio 2012 for Windows 8 or Better
2. Windows Phone 8 SDK
3. **Mac OS:**
4. MonoDevelop
5. Xamarin Studio
6. MonoMac
7. **Ubuntu:**
8. MonoDevelop
9. **CẤU TRÚC MONOGAME?**

So với XNA, MonoGame giống đến 99%, vì thế hiểu được cấu trúc XNA ta sẽ hiểu được cấu trúc làm việc của MonoGame

Mọi game trong MonoGame được bắt đầu từ class Game1.cs đây là class chính của chương trình. Nó sẽ điều khiển toàn bộ hoạt động của chương trình.

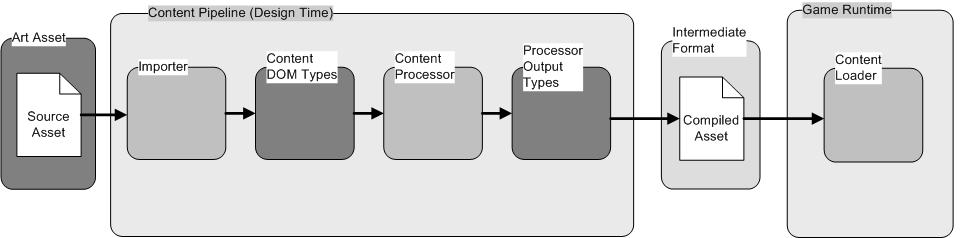
Sơ đồ hoạt động của một game:



Trong đó:

1. ***Initialize():*** khởi tạo các giá trị đầu game.
2. ***LoadConent():*** dùng để load các nội dung trong game như: Image, Font, Sound, Model,...
3. ***Update(gameTime):*** hàm dùng để xử lý nội dung, các hành động game, Input từ ngoài nhận vào, tương tác, va chạm...
4. ***Draw(gameTime):*** đây cũng là 1 hàm vô cùng quan trọng trong game, dùng để vẽ các đối tượng ra màn hình.
5. ***UnLoadContent():*** hủy tất cả các tài nguyên, dọn dẹp bộ nhớ.

PipeLine Content: nơi lưu trữ tất cả các tài nguyên của game, nó đã được mã hóa và biên dịch cùng với game. Từ đó ta có thể thông qua content mà liên kết đến những tài nguyên ta cần hiển thị:



# **Chương 2: CÁC GIẤY PHÉP SỬ DỤNG**

MonoGame được xây dựng dựa trên 4 giấy phép chính sau:

1. Giấy phép Microsoft Public License (Ms-PL).
2. Giấy phép GNU General Public License (GPL).
3. Giấy phép MIT X11 License (MIT)
4. Giấy phép Apache License.
5. **GIẤY PHÉP Ms-PL**

Giấy phép này điều chỉnh việc các phần mềm sử dụng giấy phép này, ta phải chấp nhận các điều khỏa trong giấy phép thì mới sử dụng được nó.

1. ***Định nghĩa:***

Các thuật ngữ: “reproduce” (tái sử dụng), “reproduction” (tái sản xuất), “derivative works” (các công việc tái sinh) và “distribution” (phân phối) có cùng ý nghĩa theo luật bản quyền Hoa Kỳ.

“Contribution” là một định nghĩa ám chỉ phần mềm ban đầu, hoặc những thứ được thêm vào hay làm thay đổi phần mềm.

“Contributor” là bất kỳ một người nào phân phối các “contribution” theo giấy phép này.

“Licensed patents”: những tuyên bố sáng chế của một người đóng góp “contributor” trực tiếp trên “contribution”.

1. ***Các điều khoản và hạn chế:***

(A) No Trademark License: giấy phép này không cho phép bất kỳ người phân phố nào đặt tên, sử dụng dụng logo riêng hoặc buôn bán nó.

(B) Nếu có bất kỳ tuyên bố nào ngược lại với phần mềm hoặc chống đối phần mềm thì bị xem là phạm pháp.

(C) Nếu phân phối bất kỳ bộ phận nào của phần mềm, phải giữ lại tất cả các thông tin bản quyền, bằng sáng chế, nhãn hiệu trình bày trong phần mềm.

(D) Nếu phân phối bất kỳ phần nào của phần mềm dưới dạng source code, phải tuân thủ giấy phép quy định riêng của phần mềm.

(E) Khi sử dụng, phải tự chịu rủi ro và không hề có chính sách bao hảnh phần mềm.

1. **GIẤY PHÉP GPL**

Giấy phép GPL là giấy phép công cộng và phổ biến nhất của phần mềm mở và được sử dụng cho một số thư viện phần mềm.

Giấy phép GPL phiên bản 2 bao gồm 12 điều khoản, phiên bản 3 gồm 17 điều khoản. Ý tưởng của nó là:

* Phần mềm GPL phải là phần mềm tự do, tức là người sử dụng có 4 quyền sau:
* Tự do chạy chương trình, cho bất cứ mục đích nào.
* Tự do tìm hiểu cách hoạt động của chương trình, và tự do sửa đổi nó. (Quyền truy cập mã nguồn là điều kiện tiên quyết cho quyền tự do này.)
* Tự do tái phân phối bản sao.
* Tự do cải tiến chương trình, và phát hành những gì cải tiến ra công cộng. (Quyền truy cập mã nguồn là điều kiện tiên quyết cho quyền tự do này.)
* Phần mềm phái sinh từ phần mềm GPL cũng phải là phần mềm GPL

Các giấy phép như BSD, MIT, Apache... thường không có quy định gì về phần mềm phái sinh nên xảy ra việc phần mềm phái sinh bị biến thành phần mềm độc quyền - người sử dụng bị hạn chế quyền với phần mềm phái sinh. Để đảm bảo các quyền sửa đổi, sao chép, phân phối lại với phần mềm phái sinh vẫn được trao cho người sử dụng thì Richard Stallman đã làm ra giấy phép GPL trong đó yêu cầu phần mềm phái sinh cũng phải là phần mềm GPL. Giấy phép GPL được bảo hộ bởi luật bản quyền. Giấy phép này được Stallman gọi là copyleft chơi chữ với copyright để thể hiện tinh thần đối ngược với tinh thần của copyright.

1. **GIẤY PHÉP MIT**

Giấy phép này được cấp phát miễn phí đối với bấy kỳ người dùng với một bản sao chép của phần mềm và tài liệu liên quan. Để sử dụng các dạng phần mềm này không hạn chế, không giới hạn các quyền: sử dụng, sao chép, phân phối, cấp phép hoặc bán các phiên bản phần mềm, người sử dụng phải chấp nhận các điều khoản sau đây:

* Phần mềm được cung cấp nhưng không có chế độ bảo hành, không giới hạn các bảo đảm về thương mại, phù hợp cho một mục đích cụ thể nào đó và không vi phạm trong mọi trường hợp.
* Tác giả hoặc người giữ bản quyền không phải chịu bất kỳ khiếu nại, thiệt hại hoặc các trách nhiệm khác dù trong một điều khoản của hợp đồng có ghi, các điều khoản phát sinh từ bên ngoài hay các giao dịch khác của phần mềm.

1. ASF logo**GIẤY PHÉP APACHE.**

Giấy phép Apache là một giấy phép phần mềm tự do của Quỹ phần mềm Apache (Apache Software Foundation – ASF). Giấy phép này bắt buộc phải có thông báo bản quyền khi sử dụng. Tương tự như các giấy phép phần mềm khác, Giấy phép Apache trao cho người dùng quyền tự do sử dụng phần mềm với bất kỳ mục đích nào: phân phối, chỉnh sửa hoặc phân phối bản có sửa đổi của phần mềm theo các điều khoản của giấy phép mà không lo lắng tới phí bảng quyền.

Tất cả các phần mềm do ASF hoặc các dự án và vật thể của ASF đều được cấp phép theo các điều khoản của Giấy phép Apache. Một số phần mềm không phải của ASF cũng được cấp phép theo Giấy phép Apache. Đến tháng 10 năm 2012, đã có 8708 dự án đặt tại SourceForge.net được cấp phép theo các điều khoản của Giấy phép Apache. Trong một bài viết trên blog vào tháng 5 năm 2008 Google nói rằng 25.000 trong tổng số 100.000 dự án đặt trên Google Code đang sử dụng Giấy phép Apache, trong đó có hệ điều hành Android.

* **Các điều khoản giấy phép:**

Giấy phép Apache được xem là loại giấy phép không có nhiều ràng buộc vì nó không bắt buộc [phiên bản đã thay đổi](http://vi.wikipedia.org/wiki/T%C3%A1c_ph%E1%BA%A9m_ph%C3%A1i_sinh) của phần mềm phải được phân phối với cùng giấy phép (không giống như các giấy phép [copyleft](http://vi.wikipedia.org/wiki/Copyleft)). Trong mỗi tập tin được cấp phép, bất kỳ bản quyền, bằng sáng chế, thương hiệu, và thông báo ghi công phải được giữ nguyên trong các đoạn mã khi phát hành lại (ngoại trừ các thông báo không liên quan đến tác phẩm phái sinh); và, trong tất cả các tập tin có thay đổi, phải thêm vào một thông báo nói rằng tập tin này đã được thay đổi.

Nếu có một tập tin văn bản NOTICE trong bản phân phối tác phẩm gốc, thì các tác phẩm phái sinh phải kèm theo bản sao của thông báo nào bên trong tập tin NOTICE đi kèm với tác phẩm phái sinh, bên trong mã nguồn hoặc tài liệu hướng dẫn sử dụng, hoặc bên trong một giao diện của tác phẩm phái sinh (tại nơi thường hiển thị thông báo của bên thứ ba).

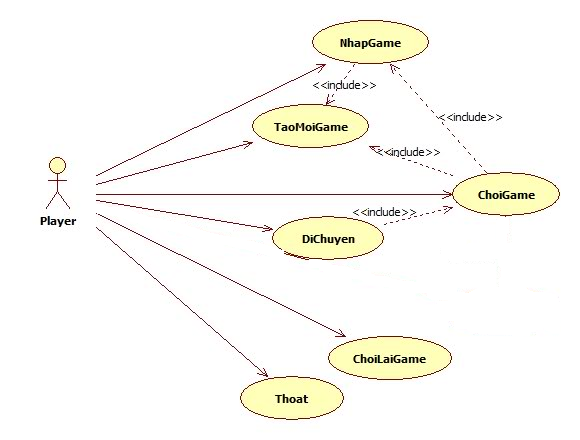
Nội dung của tập tin NOTICE không được thay đổi giấy phép, vì chúng chỉ mang tính thông tin thuần túy, và được phép thêm vào thông báo ghi công vào phần bổ sung của tập tin NOTICE, miễn là các thông báo này không bị hiểu thành thay đổi giấy phép. Những sự thay đổi có thể có các thông báo bản quyền phù hợp, và có thể có điều khoản giấy phép khác biệt cho phần sửa đổi.

Trừ khi có tuyên bố khác, bất kỳ đóng góp nào do người được cấp phép gửi cho người cấp phép sẽ tuân theo các điều khoản của giấy phép mà không kèm điều khoản và điều kiện, nhưng điều này không ảnh hưởng đến các thỏa thuận riêng rẽ giữa hai bên liên quan đến phần đóng góp này.

# **Chương 3: THIẾT KẾ HỆ THỐNG**

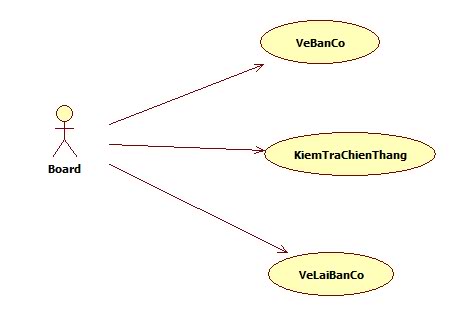
1. **MÔ HÌNH USECASE**

* **Góc độ người chơi:**



Dựa vào mô hình Usecase, ta có thể hình dung ra được các trạng thái của game:

* Khi người dùng bắt đầu truy cập vào game, có 2 trạng thái chính là NhapGame hay TaoGameMoi.
* Sau đó người dùng sẽ chuyển sang trạng thái tiếp theo là ChoiGame.
* Người dùng sẽ ở trạng thái DiChuyen khi đang ở giao diện playGame của game.
* Nếu thua hoặc thắng, kết thúc một màn, người dùng có thể chơi lại ở trạng thái ChoiLaiGame.
* Trạng thái Thoat, dùng để thoát khỏi game.
* **Góc độ lập trình game:**

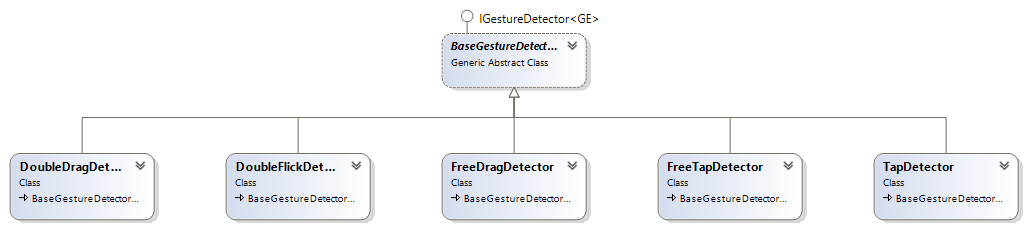


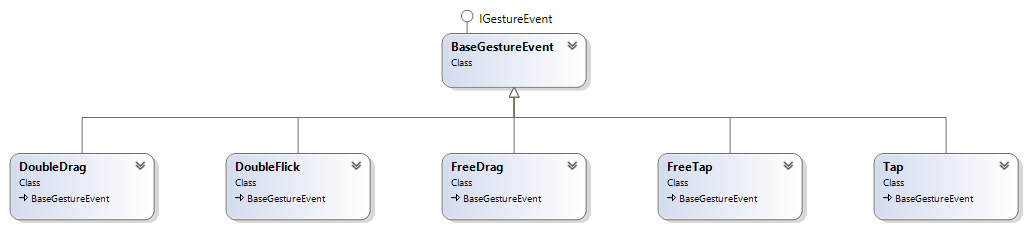
Có 2 trạng thái chính của góc độ bên trong game:

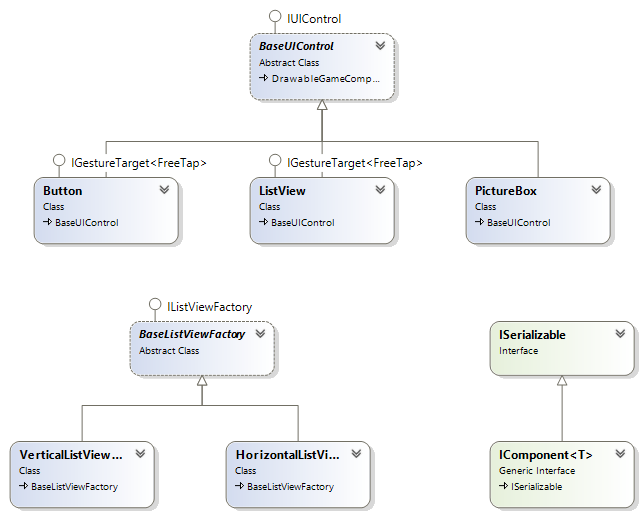
* VeBanCo: khi game bắt đầu, trạng thái này sẽ được gọi, dùng để vẽ ra bàn cờ (board) trong game cùng các đối tượng.
* KiemTraChienThang: trạng thái sẽ cập nhật liên tục trong game, kiểm tra xem người chơi thắng game hay chưa.
* VeLaiBanCo: khi trạng tái KiemTraChienThang trả về giá trị, VeLaiBanCo sẽ được gọi tùy theo giá trị trả về hoặc người dùng yêu cầu.

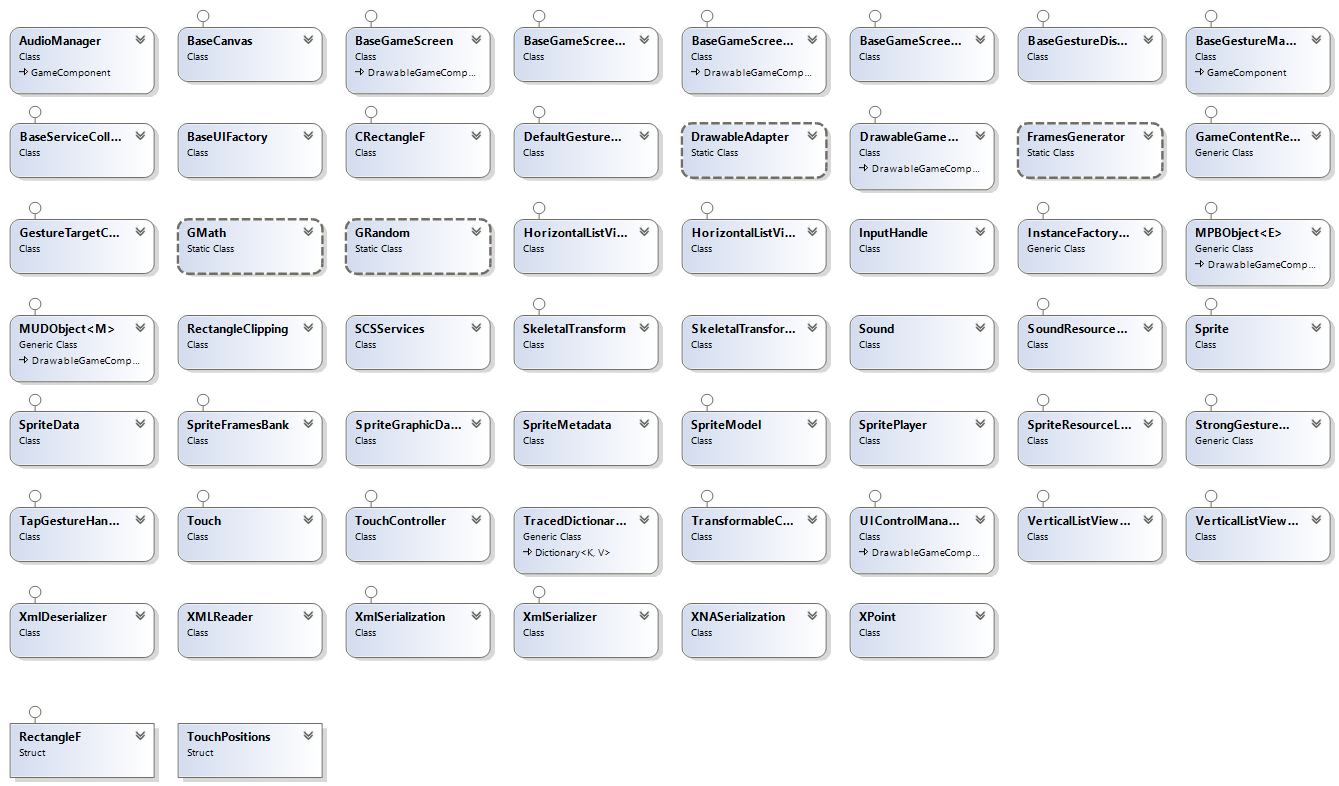
1. **MÔ HÌNH CLASS**
2. **Game Engine:** chưa các class nền tảng trong game, không chưa nội dung game

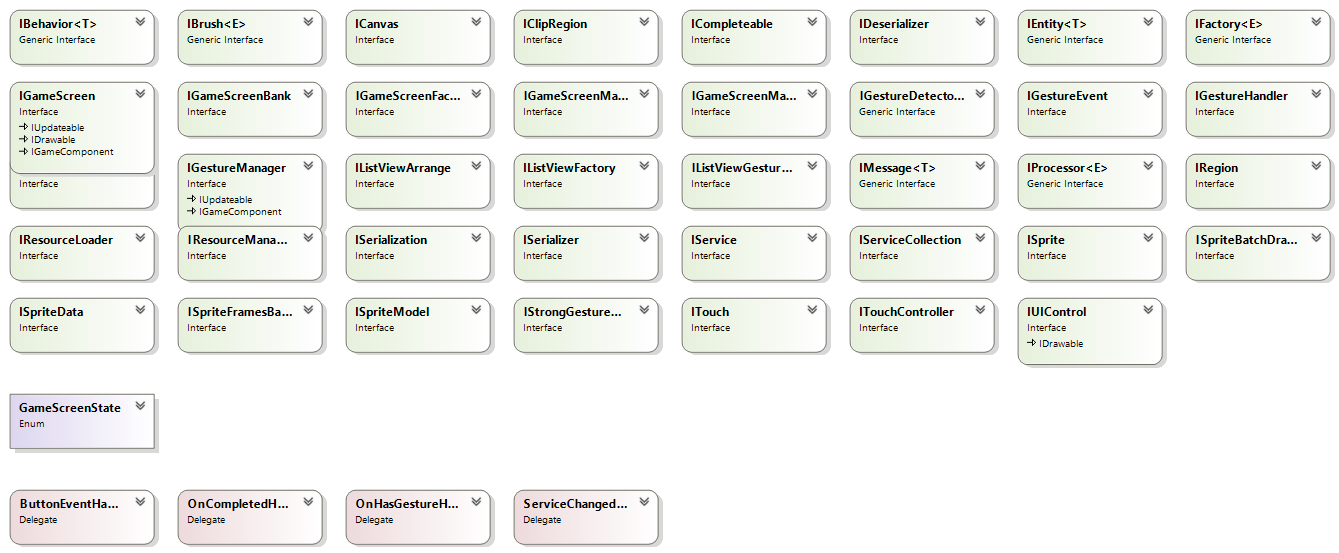
Các class quan trọng trong game:

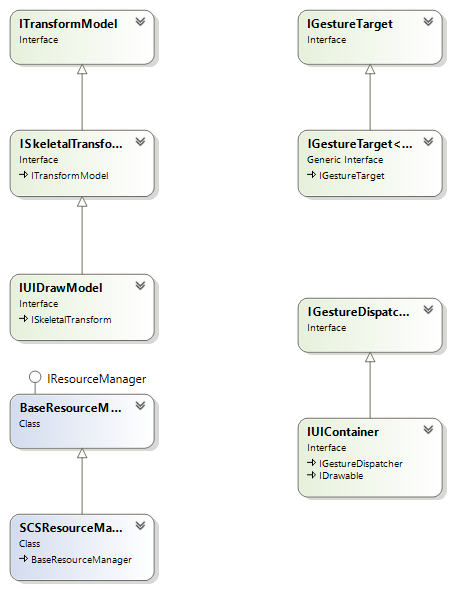




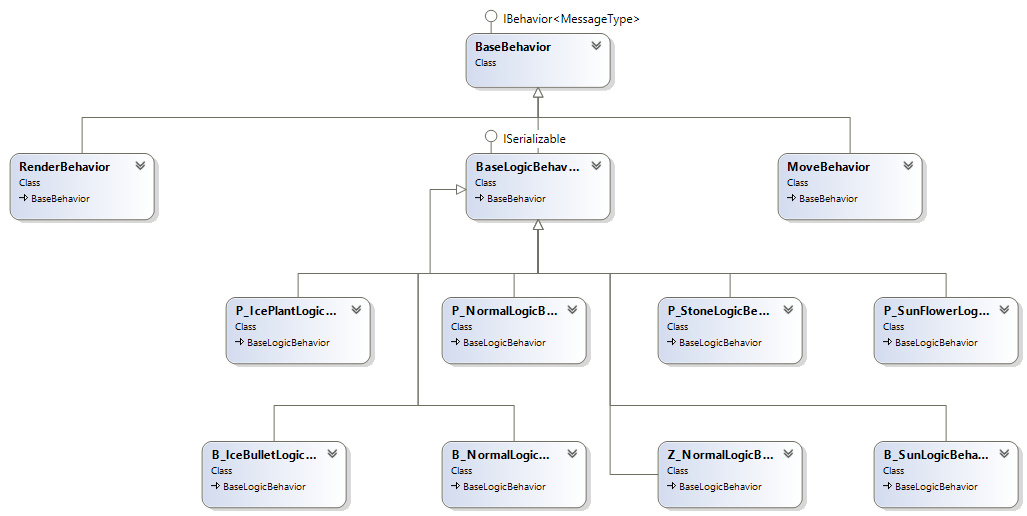


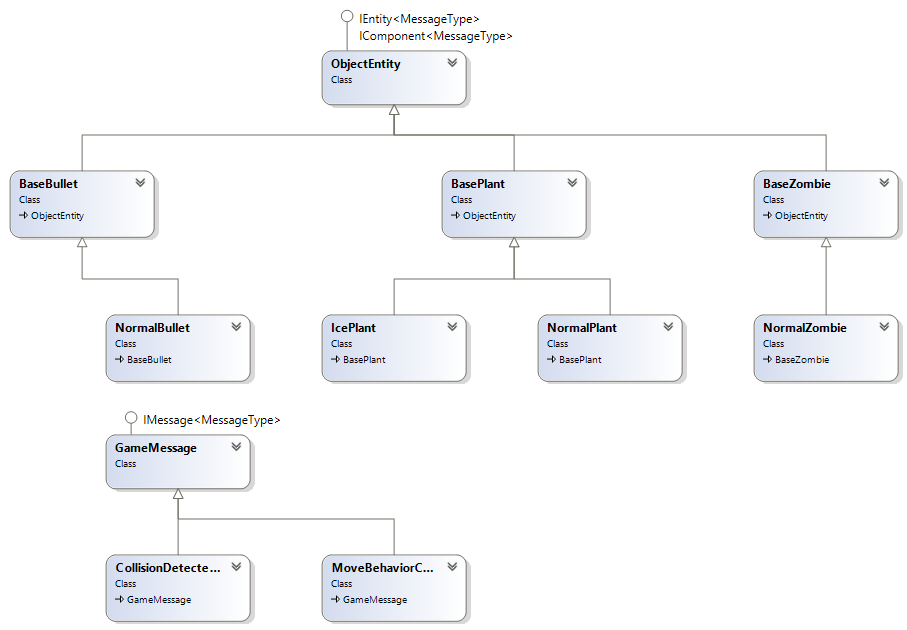


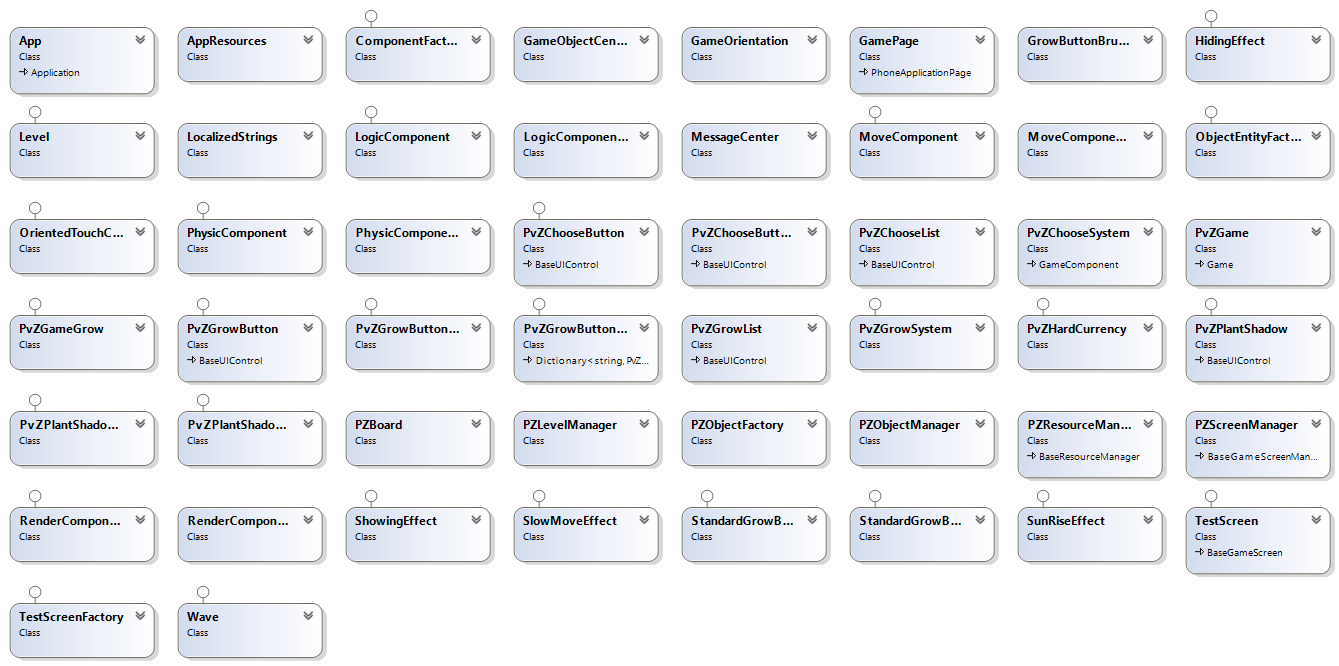




1. **PlantsVsZombies**: project chứa source code game









# **Chương 4: CÀI ĐẶT MINH HOẠT**