**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA ĐA PHƯƠNG TIỆN**

A red circle with a yellow star in the middle

Description automatically generated

**BÁO CÁO CUỐI KỲ MÔN LẬP TRÌNH GAME CƠ BẢN**

**Phát triển game “Run or Die”**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn: | Phạm Vũ Minh Tú |
| Nhóm: | 06 |
| Thành viên: | Nguyễn Quốc Đạt – B20DCPT053  Nguyễn Văn Tuấn – B20DCPT182  Nguyễn Hồng Nhung – B20DCPT150 |

**HÀ NỘI, THÁNG 01/2024**

# LỜI MỞ ĐẦU

Kính thưa thầy Phạm Vũ Minh Tú,

Trong khoảng thời gian học môn Lập trình game cơ bản, chúng em đã được phát triển kỹ năng lập trình game và ứng dụng chúng trong dự án cuối kỳ. Chân thành cảm ơn sự hướng dẫn và sự hỗ trợ nhiệt tình của thầy, chúng em đã có cơ hội tạo ra trò chơi “Run or Die” - một thước đo khả năng lập trình và sự sáng tạo của nhóm em.

Bài báo cáo này sẽ giới thiệu tổng quan về game, unity engine, phân tích và thiết kế hệ thống, cài đặt và triển khai hệ thống mà nhóm đã phát triển.

Bài báo cáo này hy vọng sẽ truyền đạt một cái nhìn sâu sắc về quá trình sáng tạo và phát triển game cũng như là điểm tựa để thảo luận và nhận xét từ phía thầy và các bạn

Mong rằng thầy và các bạn sẽ có một trải nghiệm thú vị khi khám phá trò chơi “Run or Die” thông qua bài báo cáo của nhóm chúng em.

**MỤC LỤC**

[LỜI MỞ ĐẦU i](#_Toc155644898)

[LỜI CẢM ƠN iv](#_Toc155644899)

[DANH MỤC CÁC THUẬT NGỮ VÀ CHỮ VIẾT TẮT v](#_Toc155644900)

[DANH MỤC CÁC HÌNH VẼ viii](#_Toc155644901)

[Chương 1. CƠ SỞ LÝ THUYẾT VÀ TỔNG QUAN VỀ GAME 1](#_Toc155644902)

[1.1. Tổng quan về game 1](#_Toc155644903)

[1.1.1. Giới thiệu về Video game 1](#_Toc155644904)

[1.1.2. Lịch sử phát triển game 1](#_Toc155644905)

[1.1.3. Các thể loại game 2](#_Toc155644906)

[1.2. Kiến trúc game 3](#_Toc155644907)

[1.2.1. Kiến trúc game 3](#_Toc155644908)

[1.2.2. Mô hình 3 lớp MVC 7](#_Toc155644909)

[1.2.3. Mô hình MVVM 8](#_Toc155644910)

[1.3. Quy trình phát triển game 9](#_Toc155644911)

[1.3.1. Giai đoạn tiền sản xuất 9](#_Toc155644912)

[1.3.2. Giai đoạn sản xuất 12](#_Toc155644913)

[1.3.3. Giai đoạn hậu sản xuất 13](#_Toc155644914)

[1.4. Phương pháp tiếp cận hướng đối tượng 14](#_Toc155644915)

[1.4.1. Giới thiệu về phương pháp hướng đối tượng 15](#_Toc155644916)

[1.4.2. Đặc trưng của phương pháp hướng đối tượng 15](#_Toc155644917)

[1.4.3. Lợi ích và hạn chế của phương pháp hướng đối tượng 15](#_Toc155644918)

[1.5. Tiểu kết chương 1 16](#_Toc155644919)

[Chương 2. Nghiên cứu về ứng dụng trong game 17](#_Toc155644920)

[2.1. Unity Engine 17](#_Toc155644921)

[2.1.1. Giới thiệu về Unity 17](#_Toc155644922)

[2.1.2. Lịch sử hình thành và phát triển 17](#_Toc155644923)

[2.2. Các giao diện chính và các thành phần trong Unity Engine 17](#_Toc155644924)

[2.2.1. Các giao diện chính trong Unity 17](#_Toc155644925)

[2.3.2. Các thành phần trong Unity Engine 20](#_Toc155644926)

# LỜI CẢM ƠN

Kính thưa thầy Phạm Vũ Minh Tú,

Chúng em xin gửi lời cảm ơn chân thành nhất đến thầy về sự hướng dẫn và giảng dạy của thầy trong môn Lập trình game cơ bản. Đặc biệt, chúng em muốn bày tỏ lòng biết ơn sâu sắc về dự án cuối kỳ “Run or Die” mà thầy đã hướng dẫn chúng em triển khai.

Thầy đã chia sẻ không chỉ kiến thức chuyên sâu mà còn truyền đạt tình yêu và sự đam mê của mình với lĩnh vực lập trình game. Nhờ sự hỗ trợ và tận tâm của thầy, chúng em đã có cơ hội áp dụng kiến thức đã học để tạo ra trò chơi “Run or Dỉe” với những tính năng và đặc điểm riêng biệt.

Quá trình học tập không chỉ giúp chúng em nắm bắt kiến thức mới mà còn phát triển kỹ năng làm việc nhóm, giải quyết vấn đề và quản lý dự án. Chúng em hiểu rằng những kinh nghiệm này sẽ rất hữu ích cho chúng em trong tương lai.

Một lần nữa, chúng em xin bày tỏ lòng biết ơn sâu sắc đến thầy Phạm Vũ Minh Tú vì sự đóng góp to lớn của thầy vào sự phát triển học thuật và kỹ năng của chúng em.

Trân trọng,

Nhóm 6

# DANH MỤC CÁC THUẬT NGỮ VÀ CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Từ viết tắt** | **Từ gốc tiếng Anh** | **Nghĩa tiếng việt** |
|  | Marketing | Quá trình quảng cáo, quảng bá, tiếp thị một sản phẩm hoặc dịch vụ để thu hút khách hàng và tạo sự nhận diện thương hiệu trong thị trường |
|  | Game controller | Thiết bị dùng để điều khiển hoặc tương tác với trò chơi, chẳng hạn như tay cầm gamepad, bàn phím, hoặc chuột. |
|  | Game PC | Game PC (Máy tính chơi game): Là một máy tính được thiết kế đặc biệt để chơi các trò chơi điện tử, với cấu hình phần cứng mạnh mẽ để đảm bảo hiệu suất tốt trong các trò chơi đòi hỏi. |
|  | Game console | Game Console (Máy chơi game): Là một thiết bị chơi game độc lập, thường được kết nối với TV hoặc màn hình, và đi kèm với các bộ điều khiển để chơi các trò chơi điện tử. |
|  | Game handheld | Game Handheld (Máy chơi game cầm tay): Là một thiết bị di động nhỏ gọn cho phép người chơi chơi trò chơi trên đường đi, như các máy chơi game cầm tay như Nintendo Switch hoặc Sony PlayStation Portable. |
|  | Game Arcade | Là các máy chơi game cổ điển thường được đặt tại các quán giải trí hoặc phòng trò chơi, nơi người chơi phải đưa tiền xu vào máy để chơi. |
|  | Game Casual | Các trò chơi dễ dàng tiếp cận và chơi, thường không đòi hỏi kiến thức hoặc kỹ năng sâu rộng và thường dành cho mục tiêu là người chơi giải trí và giải trí. |
|  | Game Engine | Là một hệ thống phần mềm được sử dụng để phát triển và chạy các trò chơi máy tính. Nó cung cấp các chức năng và công cụ để quản lý đồ họa, vật lý, âm thanh, cũng như quy tắc chơi trong trò chơi, giúp nhà phát triển tạo ra trải nghiệm trò chơi tốt hơn. |
| 2D | Two - Dimensional | Không gian hai chiều, thường được sử dụng để mô tả hình ảnh, đồ họa hoặc bản vẽ trên mặt phẳng, không có chiều sâu thứ ba. |
| 3D | Three - Dimensional | Không gian ba chiều, thường được sử dụng để mô tả các đối tượng, mô hình, hoặc mô phỏng có chiều sâu và thể hiện thông tin về độ cao, rộng và sâu. |
| AR/VR | Augmented Reality / Virtual Reality | Công nghệ tạo ra môi trường ảo hoàn toàn, trong đó người dùng hoàn toàn ngập trong một thế giới tạo ra bởi máy tính. AR (Augmented Reality) là công nghệ kết hợp thông tin thực tế với thông tin ảo, tạo ra một môi trường mở rộng cho người dùng. |
| AI | Artificial Intelligence | Một lĩnh vực trong khoa học máy tính và công nghệ liên quan đến việc phát triển máy tính hoặc hệ thống có khả năng học và thực hiện các nhiệm vụ thông minh giống như con người. AI có thể bao gồm học máy (machine learning), xử lý ngôn ngữ tự nhiên (natural language processing), thị giác máy tính (computer vision), và nhiều công nghệ khác để tạo ra các ứng dụng thông minh và tự động. |
| Bot |  | Một chương trình máy tính hoặc máy tự động thiết kế để thực hiện các nhiệm vụ cụ thể, thường được sử dụng để tự động hóa công việc trên Internet hoặc giao tiếp với con người thông qua chat hoặc các tương tác khác. |
| PC | Personal Computer | Ám chỉ đến máy tính cá nhân, một thiết bị tính toán dành cho sử dụng cá nhân hoặc văn phòng. |
| SDK | Software Development Kit | Là một bộ công cụ phát triển phần mềm được cung cấp cho các nhà phát triển để giúp họ phát triển ứng dụng hoặc phần mềm cho một nền tảng cụ thể, như các ứng dụng di động hoặc ứng dụng trên một hệ thống cụ thể. |

# DANH MỤC CÁC HÌNH VẼ

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Ssssssssssssssssssssssssssssssssssssssssssssssssqweqwejqwheqwjkehqw**

**Sdfdsf**

**fgfnnnnnnnnnnn**

**b**

**b**

**b**

**b**

**b**

**b**

**b**

**b**

**b**

**b**

**b**

**b**

**b**

**b**

**b**

**b**

**b**

**b**

**b**

**b**

**b**

**b**

**b**

**b**

**b**

# Chương 1. CƠ SỞ LÝ THUYẾT VÀ TỔNG QUAN VỀ GAME

## 1.1. Tổng quan về game

### 1.1.1. Giới thiệu về Video game

Video game (gọi tắt là game) là một dạng trò chơi điện tử liên quan đến tính tương tác với một giao diện người sử dụng để tạo ra một phản hồi hình ảnh trên một thiết bị hiển thị (video). Video game thường bao gồm đồ họa, âm thanh và tương tác tương tự như hình ảnh và âm thanh thời gian thực.

Một số thuật ngữ:

* Game PC: Người chơi sử dụng một máy tính cá nhân kết nối với một màn hình.
* Game console: Chơi trên một hệ máy chuyên dụng kết nối với một tivi thông thường hoặc một màn hình tích hợp các chân cắm chuyên dụng
* Game handheld (thiết bị cầm tay dùng để chơi game) chứa tất cả các linh kiện cần thiết (màn hình hiển thị, phần cứng, phần mềm lõi) để có thể mang theo và vừa với tay người sử dụng.
* Game arcade thường được hiểu là một game chơi trên một hệ máy thậm chí còn chuyên dụng hơn với các thiết bị điện tử, thiết kế chỉ để chơi một game duy nhất và người chơi sẽ chơi trong một buồng riêng biệt.

Các thiết bị đầu vào sử dụng để thao tác trong game được gọi là thiết bị điều khiển game (hay tay cầm- game controller), và thay đổi tùy theo hệ máy.

VD: Tay cầm chuyên dụng, bàn phím, chuột …

Video game cũng thường sử dụng cách khác để tương tác và cung cấp thông tin cho người chơi.

* Phổ biến nhất là âm thanh, sử dụng các thiết bị tái tạo âm thanh, như loa và tai nghe.
* Các phản hồi khác có thể là thông qua thiết bị ngoại vi có chức năng haptics: Công nghệ tạo chức năng rung phản hồi/cảnh báo cho tay cầm chơi game hay điện thoại di động.

### 1.1.2. Lịch sử phát triển game

Lịch sử phát triển game thường được chia thành nhiều giai đoạn để phản ánh sự tiến bộ và thay đổi trong ngành công nghiệp game.

Giai đoạn tiền game điện tử (1940 - 1970), thế giới đã chứng kiến sự xuất hiện của những máy tính đầu tiên, đánh dấu bước khởi đầu của ngành công nghiệp game. Trò chơi đơn giản như tic-tac-toe và các biến thể máy tính đầu tiên đã mở ra một thế giới mới của giải trí điện tử. Máy tính, trước đây chỉ là công cụ tính toán, bây giờ trở thành nền tảng cho những trò chơi sơ khai, đánh bại tính đơn điệu của công việc lập trình.

Giai đoạn game điện tử sớm (19­60 - 1970) đã chứng kiến sự xuất hiện của các trò chơi đầu tiên trên máy tính, như Spacewar trên máy tính PDP-1. Điều này đánh dấu bước ngoặt quan trọng, bắt đầu sự phát triển của ngành công nghiệp game. Các tay làm game đầu tiên bắt đầu thách thức sự sáng tạo và khám phá giới hạn của công nghệ.

Những năm 1980 – 1990 chứng kiến sự bùng nổ của thị trường trò chơi arcade với các tựa như Pac-Man và Donkey Kong. Các hệ máy console như NES, SNES và Sega Genesis được ra mắt, đưa trải nghiệm chơi game từ các phòng aracde vào tận nhà. Điều này đánh dấu sự mở rộng của ngành công nghiệp và làm thay đổi cách mọi người trải nghiệm giải trí.

Những năm 2000 chứng kiến sự phổ biến của gêm 3D với sự xuất hiện của PlayStation 2 và các nền tảng khác. Trò chơi trực tuyến đa người chơi như World of Warcraft mở ra một thế giới mới của sự tương tác và cạnh tranh trực tuyến. Sự phát triển mạnh mẽ của cả hệ thống console và PC đồng thời định hình lại cảnh đối đầu giữa các đối thủ trong ngành.

Giai đoạn 2010 – 2015 là sự phát triển mạnh mẽ của thị trường game di động với sự xuất hiện của smartphone và tablet. Cùng với đó, thực tế ảo đã bắt đầu xuất hiện, mở ra một kỷ nguyên mới trong trải nghiệm game. Người chơi không chỉ tận hưởng sự thuận tiện của di động mà còn được đắm chìm trong thế giới ảo hóa.

Trong giai đoạn hiện đại, game nhập vai và game có nội dung nghệ thuật trở thành xu hướng lớn. Người chơi không chỉ tìm kiếm trải nghiệm giải trí mà còn khám phá câu chuyện sâu sắc và nghệ thuật đẹp mắt. Game trở thành một phương tiện giải trí quan trọng, cung cấp trải nghiệm đa dạng trên nhiều nền tảng. Phát triển công nghệ đồ họa, âm thanh, và cơ sở hạ tầng mạng để hỗ trợ trải nghiệm chơi game chất lượng cao.

### 1.1.3. Các thể loại game

Có nhiều phương pháp phân loại game. Trong đó có thể phân ra những phương pháp cơ bản sau: Dựa vào cách chơi; Kỹ thuật tạo ra game; Thiết bị đầu cuối; Cách kết nối và một số phương pháp chuyên biệt khác.

Dựa vào cách thức chơi game, có các thể loại game: Hành động, bắn súng, phiêu lưu, mô phỏng cuộc sống, quản lý/ xây dựng, âm nhạc, tiệc, xếp hình, thể thao, chiến lược, mô phỏng giao thông.

Dựa vào lối chơi game cốt lõi, game casual.

* Game hành động: Thể loại game yêu cầu người chơi phải có kỹ năng và phản xạ nhanh, thường bao gồm các trò chơi bắn súng, đua xe hoặc chiến đấu. VD: Pubg Mobile, Shadow fight, Free Fire…
* Game kinh doanh: Người chơi quản lý doanh nghiệp hoặc cửa hàng để tạo ra lợi nhuận và phát triển. Thường bao gồm quản lý tài nguyên, phân phối sản phẩm và ra quyết định chiến lược. VD: Clash of Clans.
* Game thể thao: Chế độ chơi chú trọng vào các môn thể thao như bóng đá, bóng rổ, quần vợt và cử tạ. Người chơi thường điều khiển các đội thể thao hoặc vận động viên để cạnh tranh. VD: Fifa Mobile.
* Game chiến thuật: Loại game tập trung vào quản lý tài nguyên, chiến thuật, và lập kế hoạch để chiến đấu và đánh bại đối thủ. VD: Plants vs Zombies, Cờ vua…
* Game nhập vai: Người chơi đóng vai một nhân vật và theo dõi cuộc hành trình của họ trong thế giới ảo. Phát triển nhân vật, thu thập trang bị, và tham gia vào nhiều nhiệm vụ và cuộc phiêu lưu. VD: Genshin Impac

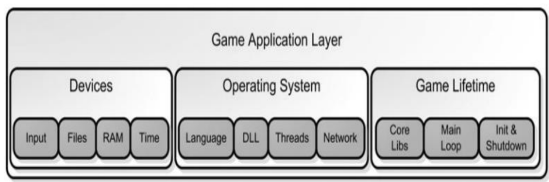
## 1.2. Kiến trúc game

### 1.2.1. Kiến trúc game

Kiến trúc game là cách mà các thành phần khác nhau trong một trò chơi máy tính được tổ chức và tương tác với nhau. Nó bao gồm cả cấu trúc tổ chức và thiết kế của trò chơi, cũng như các yếu tố kỹ thuật và công nghệ sử dụng để triển khai trò chơi đó.

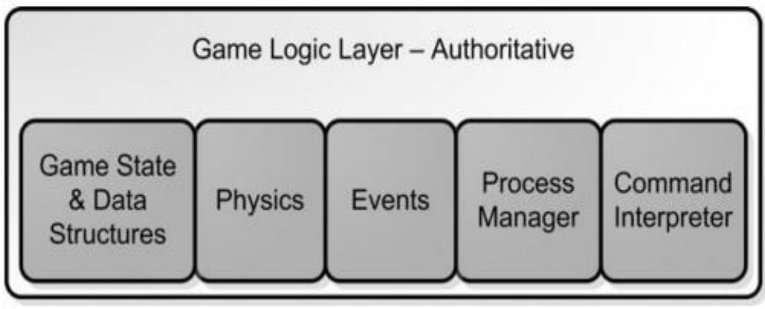
Có 3 lớp chính: Game Application layer, game logic, game view:

* Game Application layer:
* Lớp ứng dụng được chia thành các vùng khác nhau để làm việc với các thiết bị, hệ điều hành và vòng đời của game.
* Tạo và tải game logic; Tạo và đính kèm game view; Gửi yêu cầu tới CPU để thực thi chương trình.
* Chức năng đọc dữ liệu đầu vào từ thiết bị: bàn phím, chuột, máy chơi game cầm tay, bút cảm ứng, camera, bộ cảm biến, GPS,...
* Quản lý thông tin trao đổi giữa game với hệ thống file: DVD/CD/ROM, đĩa cứng, thẻ nhớ.
* Quản lý thông tin trao đổi giữa game với bộ nhớ đệm tài nguyên: Tài nguyên ko nên được tải trực tiếp từ nguồn Assets trong Project mà cần bộ đệm để tải, lưu trữ trước những tài nguyên đó → Đảm bảo tài nguyên luôn sẵn sàng được tải nhanh từ bộ đệm tài nguyên.
* Quản lý bộ nhớ: có nhiều cấu trúc dữ liệu game nhỏ và thuộc những vùng khác nhau của bộ nhớ như RAM, bộ nhớ Video…→ Hệ thống quản lý bộ nhớ cố gắng kiểm soát tất cả các vùng dữ liệu đó.
* Quản lý việc khởi tạo, thoát, vòng lặp game:
  + Thực thi:
    - Ứng dụng: Khi có tác động từ người dùng.
    - Game: Khi không có điều khiển đầu vào từ người chơi.
  + Vòng lặp game: nhận điều khiển đầu vào theo thứ tự -> Gửi yêu cầu đến game logic → Thể hiện trạng thái game đến các giao diện game.



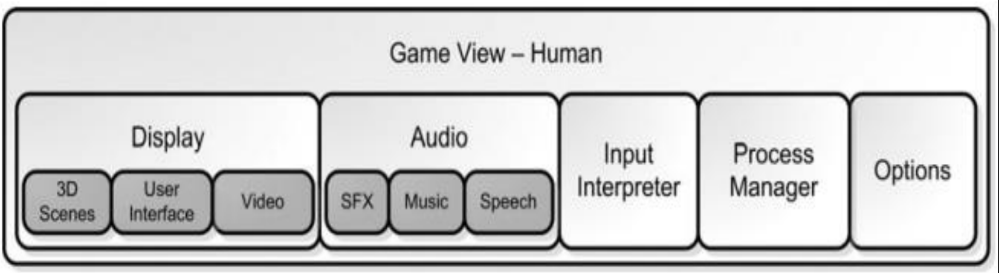
Hình 1.1. Game application layer

* Một số mã nguồn khác trong lớp ứng dụng game
  + Hệ thống đồng hồ : Đồng bộ các hiệu ứng âm thanh, hình ảnh, di chuyển..
  + Xử lý chuỗi: Chuyển đổi ngôn ngữ trong game.
  + Thư viện tải động (DLLs).
  + Luồng và đồng bộ luồng: đảm bảo việc tải tài nguyên nhanh và nhất quán.
  + Giao tiếp mạng.
  + Khởi tạo.
  + Vòng lặp chính.
  + Tắt chương trình.
* Game Logic layer:
* Lớp logic game định nghĩa toàn bộ thế giới quan của game: trạng thái của game, những đối tượng có trong game và tương tác giữa chúng với nhau.



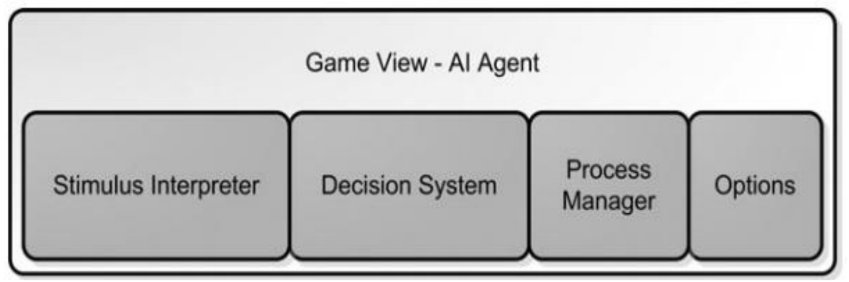
Hình 1.2. Game logic layer

* Quản lý trạng thái game và cấu trúc dữ liệu trong game: Dữ liệu trong cấu trúc dữ liệu của các đối tượng sẽ được game logic đọc và trả về lớp giao diện game, thể hiện trạng thái của game.
* Quản lý đặc tính vật lý và xử lý va chạm của các đối tượng trong game
* Một số đặc tính vật lý của các đối tượng: trọng lực, gia tốc, vận tốc,góc quay...
* Hệ thống vật lý cũng sẽ thực hiện các xử lý va chạm của các đối tượng trong game, các đối tượng sẽ chuyển động, va chạm và ảnh hưởng nhau như nào.
* Quản lý sự kiện trong game
  + Khi game logic thay đổi trạng thái game (ví dụ như di chuyển nhân vật, va chạm nhân vật,...) thì một loạt các hệ thống con sẽ có phản hồi lại.
  + Hệ thống quản lý sự kiện như một chất keo dính kết nối các sub-systems trong hệ thống lại với nhau.
* Quản lý tiến trình:
* Trong game nhiều tiến trình song song thực hiện
* Quản lý tiến trình trong game để đảm bảo các tiến trình không xung đột.
* Command Interpreter: Thực hiện các lệnh đầu vào từ người chơi , trả về trạng thái game thay đổi hiển thị lên giao diện game.
* Game View Layer:
* Góc nhìn người chơi:
* Giao diện game là một tập hợp hệ thống tương tác với game logic để hiển thị giao diện game với từng vai game .
* Vai game: người chơi / đối tượng AI.



Hình 1.3. Game view – human

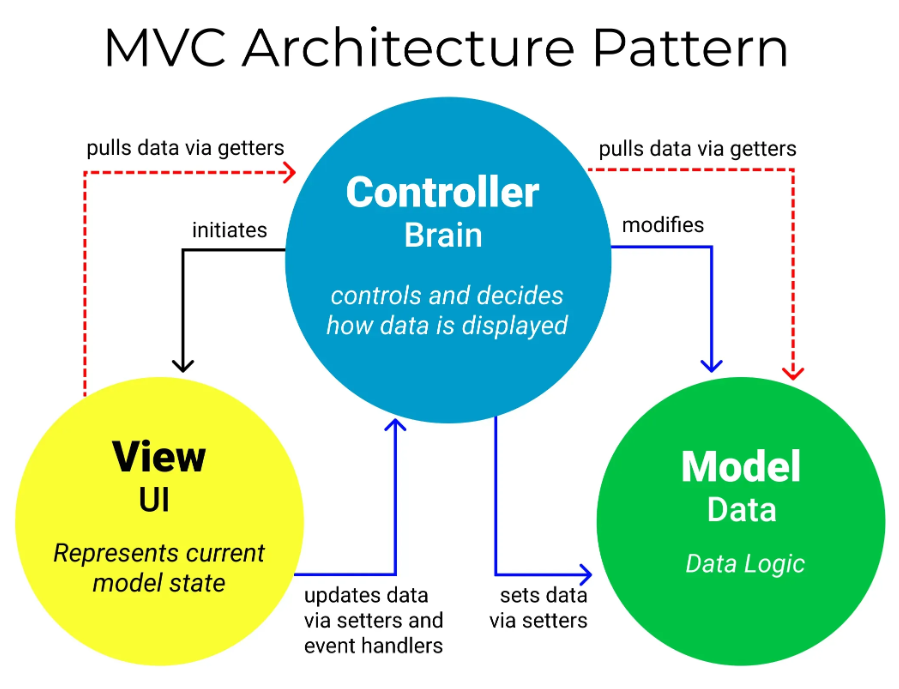
* Game display
  + - Vẽ các đối tượng trong màn hình game.
    - Hiển thị luồng video, giao diện người chơi.
* Game audio
  + Chia làm 3 lĩnh vực chính: hiệu ứng âm thanh, nhạc, tốc độ.
  + Hiệu ứng âm thanh có thể được xử lý một cách dễ dàng trong game, đơn giản chỉ cần tải tệp tin nhạc đuôi WAV lên và gửi nó tới DirectX với các tham số về âm lượng và vòng lặp.
  + Tốc độ âm thanh là một vấn đề phức tạp hơn nhiều, nó sẽ theo dõi tất cả các bits âm thanh được ghi và khớp chúng đồng bộ với nhân vật trong môi trường.
* Hiển thị giao diện người chơi: Mỗi khi trạng thái game thay đổi, giao diện game sẽ thay đổi theo.
* Quản lý tiến trình trong giao diện game: Quản lý việc hiển thị (hiệu ứng hình ảnh, âm thanh, video...) trên giao diện game.
* Quản lý tùy chọn trong game: Hầu hết các game sẽ có các lựa chọn cấu hình như âm lượng hiệu ứng âm thanh, cấu hình tỉ lệ màn hình hiển thị,...
* Góc nhìn AI:
* Hành vi của đối tượng AI cũng giống như hành vi của con người.
* Điều khiển hành vi của AI với Stimulus Interpreter:
* Stimulus interpreter nhận các sự kiện giống như tất cả các game view khác như sự di chuyển của đối tượng, va chạm, ....
* Người lập trình AI sẽ lập trình cách mà AI phản ứng lại khi nhận các sự kiện. Ví dụ, AI sẽ trả về các âm thanh tương ứng khi có các sự kiện trong game diễn ra như va chạm, nhân vật nhảy nhót hay chết.
* Tất cả các hành vi của AI sẽ được điều khiển bởi Stimulus Interpreter.
* Decision system: Hệ thống này được tùy chỉnh để biên dịch các tín hiệu từ Stimulus Interpreter thành các hành động
* Process manager: Nếu AI cần giải quyết một số vấn đề khó ví dụ như làm thế nào để điều hướng sang một môi trường phức tạp hoặc tạo một bước đi tiếp theo trong ván cờ, cần một AI Manager để quản lý. AI manager có thể quản lý và mở rộng các chức năng của AI.
* Options: Danh sách tùy chọn các khả năng của AI để tùy chỉnh thông qua 1 tệp tin .text.



Hình 1.4. Game view – AI

### 1.2.2. Mô hình 3 lớp MVC

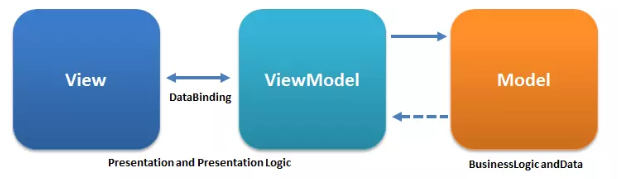
Mô hình 3 lớp MVC là một dạng kiến trúc ứng dụng, trong đó các phần của chương trình được gom vào 3 lớp khác nhau: Model, View, Controller.



Hình 1.5. Mô hình 3 lớp MVC

* **Model:** Là bộ phận có chức năng **lưu trữ toàn bộ dữ liệu** của ứng dụng và là cầu nối giữa 2 thành phần bên dưới là **View** và **Controller**. Một model là dữ liệu được sử dụng bởi chương trình. Đây có thể là cơ sở dữ liệu, hoặc file XML bình thường hay một đối tượng đơn giản. Chẳng hạn như biểu tượng hay là một nhân vật trong game.
* View: Đây là phần giao diện (theme) dành cho người sử dụng. View là phương tiện hiển thị các đối tượng trong một ứng dụng. Chẳng hạn như hiển thị một cửa sổ, nút hay văn bản trong một cửa sổ khác. Nó bao gồm bất cứ thứ gì mà người dùng có thể nhìn thấy được.
* Controller: Là bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua View. Một controller bao gồm cả Model lẫn View. Nó nhận input và thực hiện các update tương ứng.

### 1.2.3. Mô hình MVVM

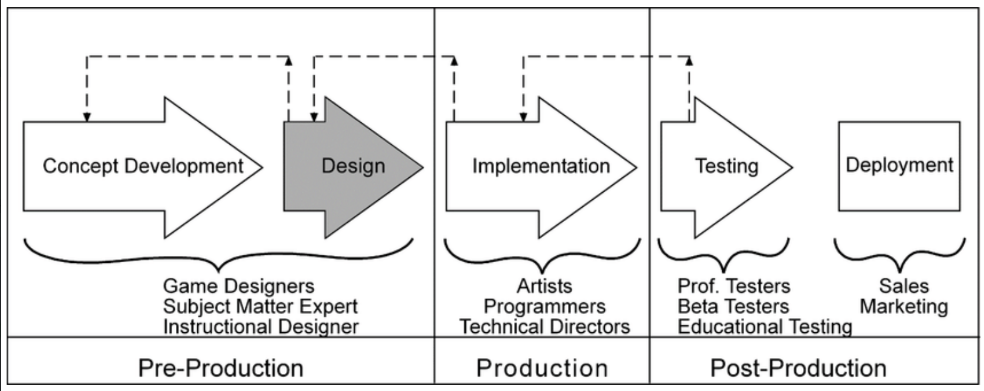
****

Hình 1.6. Mô hình MVVM

Trong mô hình MVVM, các tầng bên dưới sẽ không biết được các thông tin gì về tầng bên trên nó. ViewModel không hề biết gì về View, một ViewModel có thể được sử dụng cho nhiều View (one-to-many). ViewModel sử dụng Observer design pattern để liên lạc với View (thường được gọi là binding data, có thể là 1 chiều hoặc 2 chiều tùy nhu cầu của ứng dụng).

* View: Thành phần giao diện của ứng dụng. Tương tự như ở trong mô hình MVC, View là thành phần duy nhất mà người dùng có thể tương tác được trong chương trình, nó chính là thành phần mô tả dữ liệu.
* Model: Cũng tương tự như trong mô hình MVC. Model là các đối tượng giúp truy xuất và thao tác trên dữ liệu thực sự.
* ViewModel: Sẽ đảm nhận công việc đồng bộ dữ liệu từ model lên View. Mối quan hệ giữa View và View-Model là View sẽ được ánh xạ tới View-Model nhưng View-Model lại không biết thông tin gì về View. Nó được ẩn dấu qua cách sử dụng Data-binding và cơ chế của mô hình Observer. Một View-Model có thể được ánh xạ từ nhiều View.

## 1.3. Quy trình phát triển game



Hình 1.7. Quy trình phát triển trò chơi

Quá trình phát triển trò chơi thông thường được chia thành ba giai đoạn chính: tiền sản xuất (pre-production), sản xuất (production), và sau sản xuất (post-production). Trong giai đoạn tiền sản xuất, đội ngũ phát triển tập trung vào việc xây dựng cơ sở hạ tầng cho dự án. Điều này bao gồm việc xây dựng ý tưởng cơ bản (concept) và thiết kế (design) của trò chơi. Trong giai đoạn này, quyết định về cốt truyện, gameplay, và ý tưởng nghệ thuật quyết định hình dạng tổng thể của trò chơi.

Tiếp theo là giai đoạn sản xuất, nơi các ý tưởng được chuyển đổi thành hiện thực thông qua quá trình triển khai (implementation). Nhóm phát triển tập trung vào việc xây dựng và kết hợp các yếu tố khác nhau như đồ họa, âm thanh, và cơ chế gameplay. Giai đoạn này đòi hỏi sự hợp tác chặt chẽ giữa các thành viên của nhóm để đảm bảo rằng mọi phần của trò chơi hoạt động mượt mà và phù hợp với ý tưởng ban đầu.

Cuối cùng, giai đoạn sau sản xuất tập trung vào thử nghiệm (testing) và triển khai (deployment). Đội ngũ thử nghiệm đảm bảo rằng trò chơi không có lỗi và đáp ứng đúng mong đợi của người chơi. Sau khi hoàn thiện, trò chơi được triển khai ra thị trường, sẵn sàng để người chơi trải nghiệm. Quá trình này đòi hỏi sự chăm chỉ, sáng tạo và kiểm soát chất lượng để tạo ra một sản phẩm game thành công.

### 1.3.1. Giai đoạn tiền sản xuất

* Idea: Ngắn gọn, hấp dẫn về ý tưởng cốt truyện, chủ đề hoặc điểm độc đáo của trò chơi. Ý tưởng cần nhanh chóng truyền đạt các tính năng chính làm cho trò chơi nổi bật so với những trò chơi khác trên thị trường. Đây là 1 bước quan trọng, không chỉ giúp xác định bản chất của trò chơi mà còn phục vụ như một nguyên tắc hướng dẫn cho đội ngũ phát triển trong suốt quá trình tạo ra trò chơi.
* Concept: Là ý tưởng cốt, khái quát hoặc mô tả tổng quan về trò chơi. Là nền tảng ban đầu cho quy trình phát triển, cung cấp 1 cái nhìn tổng quan về những gì trò chơi đem lại. Nó giúp định hình cốt truyện, cơ chế chơi và các yếu tố thiết kế quan trọng khác.
* Cốt truyện và bối cảnh: Mô tả cốt truyện chính và nền tảng bối cảnh của trò chơi. Điều này có thể bao gồm thế giới trong trò chơi, thời gian, và tình huống khởi đầu.
* Mục tiêu và nhiệm vụ: Định rõ mục tiêu chơi và nhiệm vụ mà người chơi phải thực hiện. Điều này có thể liên quan đến việc giải quyết câu đố, chiến đấu, khám phá, xây dựng, hoặc bất kỳ hoạt động nào khác.
* Cơ chế chơi: Mô tả cách người chơi tương tác với trò chơi. Bao gồm các yếu tố như cách di chuyển, tương tác với đối tượng, hệ thống chiến đấu, cơ chế thao tác, và nhiều hơn nữa.
* Nhân vật: Đặc tả những nhân vật chính và phụ trong trò chơi, bao gồm ngoại hình, tính cách, kỹ năng và mục tiêu cá nhân.
* Thế giới game: Trình bày thông tin về thế giới mà trò chơi diễn ra, bao gồm địa điểm, văn hóa, lịch sử và các yếu tố tạo nên bối cảnh cho trò chơi.
* Phong cách nghệ thuật: Xác định hướng nghệ thuật và thiết kế hình ảnh cho trò chơi, bao gồm loại hình nghệ thuật (2D, 3D), màu sắc, hình vẽ, và phong cách trực quan.
* Âm nhạc và Âm thanh: Xác định cách âm nhạc và hiệu ứng âm thanh sẽ tương tác với trò chơi, tạo nên không gian âm thanh độc đáo và phù hợp với bối cảnh.
* Tính độc đáo của giá trị: Liệt kê những điểm độc đáo và giá trị đặc biệt mà trò chơi đem lại so với các trò chơi khác.
* Pitch:
* Thuật ngữ thường được sử dụng trong ngành công nghiệp giải trí, đặc biệt là trong việc tiếp thị và quảng bá các ý tưởng, dự án, sản phẩm hoặc dịch vụ.
* "Pitch" có thể ám chỉ việc trình bày một ý tưởng cốt truyện, cơ chế chơi hoặc khái niệm của sản phẩm đó trước một nhóm nhà đầu tư, đối tác hoặc người quản lý dự án.
* Mục tiêu của một pitch là thuyết phục người nghe rằng ý tưởng đó có giá trị và tiềm năng thành công.
* Game Design Document (GDD): Là một tài liệu chi tiết mô tả toàn bộ cấu trúc, yếu tố thiết kế, cơ chế chơi và các khía cạnh khác của một trò chơi video. GDD là tài liệu hướng dẫn cho đội phát triển và là cơ sở để hiểu và thực hiện các phần khác nhau của trò chơi.

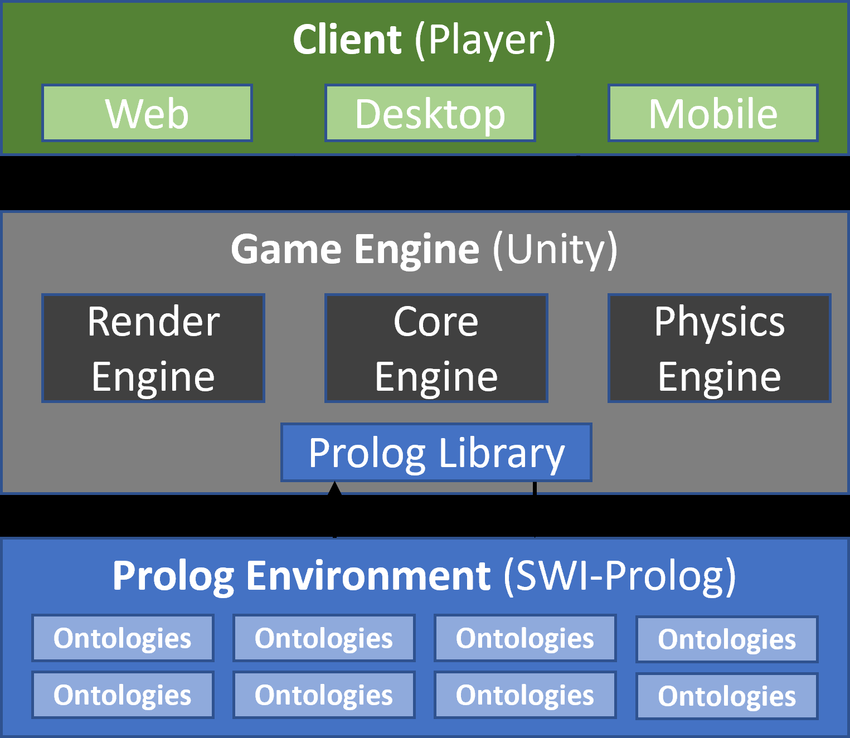


Hình 1.8. Tài liệu thiết kế trò chơi

* Prototype: "Prototype" (nguyên mẫu) là một phiên bản ban đầu, thô sơ và thường chưa hoàn chỉnh của một sản phẩm hoặc dự án, được tạo ra nhằm kiểm tra và thử nghiệm các ý tưởng, cơ chế hoặc tính năng cốt lõi. Trong ngành công nghiệp trò chơi video, một prototype game thường là phiên bản ban đầu của trò chơi, được phát triển để kiểm tra và hiểu rõ hơn về cách trò chơi hoạt động trong thực tế.

### 1.3.2. Giai đoạn sản xuất

* Thiết kế đồ họa (Graphic Design): Trong phát triển trò chơi video là quá trình tạo ra các yếu tố nghệ thuật và hình ảnh để thể hiện trực quan cho trò chơi. Điều này bao gồm thiết kế các nhân vật, môi trường, giao diện người dùng, hình ảnh nền, biểu tượng và nhiều phần nghệ thuật khác để tạo nên trải nghiệm hấp dẫn cho người chơi.
* Thiết kế âm thanh (Sound Design) trong phát triển trò chơi video là quá trình tạo ra các hiệu ứng âm thanh, âm nhạc và các yếu tố âm thanh khác để thể hiện và nâng cao trải nghiệm âm thanh của trò chơi. Thiết kế âm thanh đóng vai trò quan trọng trong việc tạo ra không gian âm thanh độc đáo, tạo cảm xúc và tương tác thú vị cho người chơi.
* Lập trình tích hợp (Integration Programming) trong ngành trò chơi video ám chỉ việc kết hợp các thành phần riêng lẻ của trò chơi thành một hệ thống hoạt động tương thích và mượt mà. Điều này bao gồm việc viết mã và lập trình để đảm bảo rằng các yếu tố như cơ chế chơi, hệ thống, nội dung và yếu tố nghệ thuật khác hoạt động cùng nhau như dự kiến.



Hình 1.9. Lập trình tích hợp

* Cơ chế chơi: Lập trình để đảm bảo rằng các cơ chế chơi như di chuyển, tương tác với đối tượng, hệ thống chiến đấu và nhiều cơ chế khác hoạt động một cách chính xác và mượt mà.
* Giao diện người dùng: Lập trình giao diện để đảm bảo rằng người chơi có thể tương tác với trò chơi qua các menu, nút bấm, hộp thoại và giao diện khác.
* Kết nối dữ liệu: Lập trình để kết nối dữ liệu từ các nguồn khác nhau như cơ sở dữ liệu, tệp tin lưu trữ và mạng để đảm bảo rằng trò chơi có thể lấy và lưu trữ thông tin cần thiết.
* Quản lý tài nguyên: Lập trình để quản lý tài nguyên như mô hình 3D, hình ảnh, âm thanh và nhiều tài nguyên khác để tối ưu hóa hiệu suất và tải dữ liệu hiệu quả.
* Xử lý sự kiện: Lập trình xử lý các sự kiện trong trò chơi như tương tác người chơi, thời gian, va chạm và các sự kiện khác để tạo ra phản ứng thích hợp.
* Thử nghiệm và sửa lỗi: Lập trình để thực hiện kiểm tra và sửa lỗi để đảm bảo rằng trò chơi hoạt động mượt mà và ổn định.
* Tối ưu hóa hiệu suất: Lập trình để tối ưu hóa hiệu suất trò chơi, đảm bảo rằng nó chạy mượt mà trên các nền tảng và thiết bị khác nhau.
* Kết hợp nghệ thuật và âm thanh: Lập trình để tích hợp yếu tố nghệ thuật và âm thanh, đảm bảo rằng trải nghiệm âm thanh và hình ảnh hoạt động tương thích và tạo cảm giác đồng nhất cho người chơi.

### 1.3.3. Giai đoạn hậu sản xuất

* Kiểm thử và bảo trì (Testing and Maintenance) là giai đoạn quan trọng trong quy trình phát triển trò chơi video, nhằm đảm bảo rằng trò chơi hoạt động một cách chính xác, mượt mà và đáp ứng được mong đợi của người chơi. Giai đoạn này bao gồm việc kiểm tra, phát hiện và sửa lỗi, cũng như duy trì và cải tiến trò chơi sau khi đã phát hành.
* Kiểm thử chất lượng: Thực hiện kiểm thử chéo và kiểm tra chất lượng để tìm lỗi, bug và vấn đề kỹ thuật khác. Điều này bao gồm kiểm tra các tương tác, cơ chế chơi, hệ thống và nhiều khía cạnh khác của trò chơi.
* Kiểm tra chức năng: Đảm bảo rằng tất cả các tính năng và cơ chế chơi hoạt động đúng cách và theo đúng thiết kế.
* Kiểm tra hiệu suất: Đảm bảo rằng trò chơi chạy mượt mà và đáp ứng hiệu suất tốt trên các nền tảng và thiết bị khác nhau.
* Kiểm tra tương thích: Kiểm tra trò chơi trên nhiều thiết bị và nền tảng khác nhau để đảm bảo rằng nó hoạt động một cách đáng tin cậy trên tất cả các môi trường này.
* Kiểm tra bảo mật: Đảm bảo tính bảo mật của trò chơi bằng cách kiểm tra và ngăn chặn các lỗ hổng bảo mật có thể bị lợi dụng.
* Sửa lỗi và Cải tiến: Xử lý và sửa lỗi, bug và vấn đề kỹ thuật được tìm thấy trong quá trình kiểm thử. Đồng thời, cải tiến trò chơi dựa trên phản hồi từ người chơi.
* Duy trì và Cập nhật: Tiếp tục cung cấp hỗ trợ, sửa lỗi và phát hành các bản cập nhật sau khi trò chơi đã được phát hành để duy trì sự hấp dẫn và chất lượng của trò chơi.
* Tối ưu hóa: Tiếp tục tối ưu hóa hiệu suất, tải dữ liệu và các khía cạnh kỹ thuật khác để đảm bảo trò chơi hoạt động tốt và mượt mà.



Hình 1.10. Kiểm thử và bảo trì

* Marketing(Tiếp thị) trong ngành trò chơi video là quá trình xây dựng và thực hiện các chiến lược để quảng bá, quảng cáo và tiếp cận đối tượng mục tiêu, nhằm tạo sự nhận biết, tạo sự tò mò và tăng doanh số bán hàng cho trò chơi. Marketing đóng vai trò quan trọng trong việc tạo sự chú ý và thu hút người chơi đến trò chơi của bạn.

## 1.4. Phương pháp tiếp cận hướng đối tượng

### 1.4.1. Giới thiệu về phương pháp hướng đối tượng

Phương pháp hướng đối tượng (Object Oriented Programming) là một mô hình lập trình được lập trình được thiết kế dựa trên khái niệm về “đối tượng”. Trong lập trình hướng đối tượng, chương trình được tổ chức thành các đối tượng, mỗi đối tượng có trạng thái (dữ liệu) và các phương thức (hành vi) để thao tác với trạng thái đó. Mô hình này hỗ trợ việc phát triển, duy trì và mở rộng chương trình 1 cách dễ dàng.

### 1.4.2. Đặc trưng của phương pháp hướng đối tượng

**Tính đóng gói (Encapsulation)**: Tính chất này không cho phép người dùng trực tiếp tác động đến dữ liệu bên trong đối tượng mà phải thông qua các phương thức mà đối tượng cung cấp. Tính chất này đảm bảo tính toàn vẹn của đối tượng.

**Tính kế thừa (Inheritance)**: Kế thừa, tái sử dụng phương thức, thuộc tính của lớp cơ sở và lớp kế thừa được gọi là lớp con, nó sẽ thừa hưởng những gì lớp cha có và cho phép.

**Tính đa hình (Polymorphism)**: Tính đa hình cho phép các chức năng (method) khác nhau được thực thi khác nhau trên các đối tượng khác nhau.

**Tính trừu tượng(Abstraction)**: Tập trung vào cốt lõi của đối tượng, bỏ qua những thứ không liên quan và không quan trọng.

### 1.4.3. Lợi ích và hạn chế của phương pháp hướng đối tượng

**Lợi ích:**

* Tính tái sử dụng: Có thể sử dụng lại các đối tượng và lớp đã được xây dựng trong các dự án khác.
* Tính Mô-đun hóa: Chia chương trình thành các module nhỏ, giúp quản lý và duy trì dễ dàng.
* Phát triển nhanh chóng: Có thể phát triển các phần của chương trình 1 cách độc lập và kết hợp chúng sau đó.
* Dễ dàng hiểu và duy trì: Mô hình này sẽ giúp tạo ra một cấu trúc tổ chức rõ ràng và dễ hiểu.
* Tính mở rộng: Dễ dàng mở rộng hệ thống bằng cách thêm vào các đối tượng và các lớp mới.

**Hạn chế:**

* Khó khăn trong quá trình học: Có thể đòi hỏi 1 thời gian để hiểu và làm chủ kiến thức về hướng đối tượng, đặc biệt với người mới học lập trình.
* Phức tạp: Dự án lớn có thể trở nên phức tạp nếu không quản lý tốt cấu trúc hướng đối tượng.
* Hiệu suất: Có thể có một số chi phí hiệu suất do việc sử dụng nhiều đối tượng và lớp.

## 1.5. Tiểu kết chương 1

Như vậy qua chương 1, các khái niệm về video game, các kiến thức liên quan tới kiến trúc ứng dụng theo mô hình 3 lớp MVC, MVVM và kiến trúc video game được trình bày rất chi tiết. Kiến trúc video game bao gồm 3 lớp: Lớp giao diện game, lớp logic và lớp ứng dụng: Lớp ứng dụng làm việc với các thiết bị, hệ điều hành và vòng đời của game. Lớp giao diện thể hiện khung nhìn game với từng vai game. Lớp logic xử lý các vấn đề diễn ra trong game, lớp chung chuyển giữa 2 lớp ứng dụng và lớp giao diện game. Bên cạnh đó đã trình bày các kiến thức liên quan tới quá trình sản xuất game gồm các giai đoạn tiền sản xuất, sản xuất và hậu sản xuất, và phương pháp tiếp cận hướng đối tượng rất phổ biến trong quy trình sản xuất game…

# Chương 2. Nghiên cứu về ứng dụng trong game

## 2.1. Unity Engine

### 2.1.1. Giới thiệu về Unity

Unity là một “cross-platform game engine” tạm hiểu là công cụ phát triển game đa nền tảng được phát triển bởi Unity Technologies. Game engine này được sử dụng để phát trển game trên PC, consoles, thiết bị di động và trên websites.

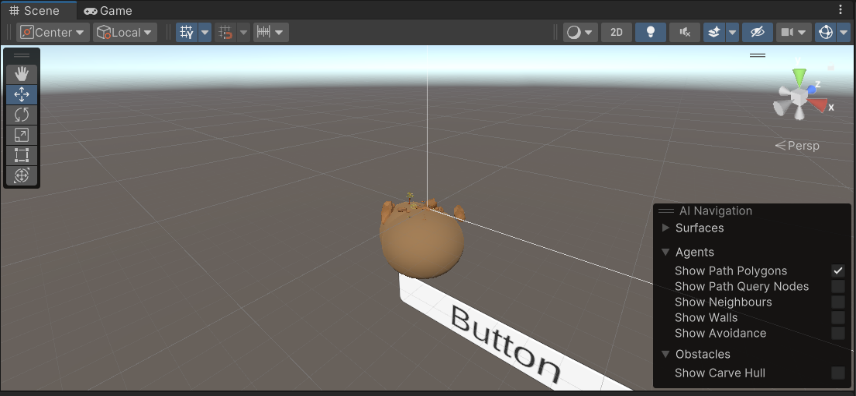
### 2.1.2. Lịch sử hình thành và phát triển

Unity là một game engine [đa nền tảng](https://vi.wikipedia.org/wiki/%C4%90a_n%E1%BB%81n_t%E1%BA%A3ng) được phát triển bởi Unity Technologies, chủ yếu để phát triển video game cho máy tính, consoles và điện thoại. Lần đầu tiên nó được công bố chạy trên hệ điều hành OS X, tại Apple's Worldwide Developers Conference vào năm 2005. Hơn [50% số lượng game trên thị trường](https://www.gamedeveloper.com/business/game-engines-on-steam-the-definitive-breakdown) được sản xuất bởi Unity. Một vài tựa game vô cùng nổi tiếng được tạo ra bởi Unity có thể được kể đến như Pokémon Go, Hearthstone, Ori And The Blind Forest, Monument Valley, Axie Infinity,… Độ “phủ sóng” của Unity rất rộng, có thể được áp dụng phổ biến trong nhiều dòng game khác nhau từ game “hạng nặng” Triple A (AAA) cho đến game giáo dục đơn giản cho con nít.

## 2.2. Các giao diện chính và các thành phần trong Unity Engine

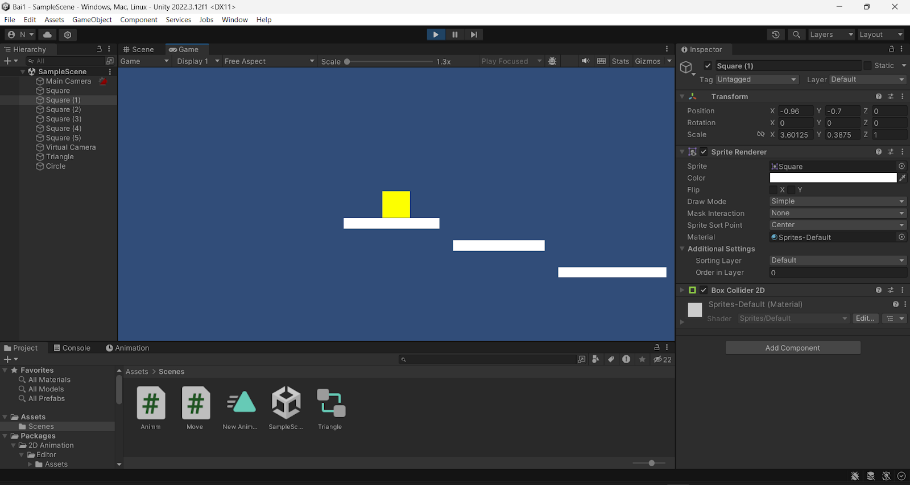
### 2.2.1. Các giao diện chính trong Unity

Scene View (Khung Hiển Thị Khung Cảnh): Đây là nơi bạn xây dựng và chỉnh sửa các khung cảnh (scenes) trong trò chơi. Ta có thể thực hiện di chuyển cửa sổ, các thao tác cơ bản trên đối tượng như xoay nhân vật theo các trục, di chuyển nhân vật, thay đổi tỉ lệ, đổi góc nhìn, … Có thể sử dụng bằng các phím tắt như Q, W, E, R tương ứng với quay bàn tay, di chuyển đối tượng, xoay đối tượng, chỉnh tỉ lệ đối tượng.



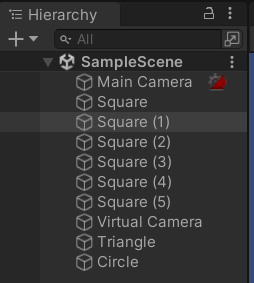
Hình 2.1. Giao diện hiển thị khung cảnh trong Unity

Game View (Khung Hiển Thị Trò Chơi): Khung này cho phép xem trước trò chơi trong thời gian thực. Có thể kiểm tra cách trò chơi sẽ hiển thị cho người chơi, cho phép xem trước và kiểm tra trò chơi trong quá trình phát triển, từ đó phát hiện và sửa lỗi cũng như đánh giá trải nghiệm của người chơi.



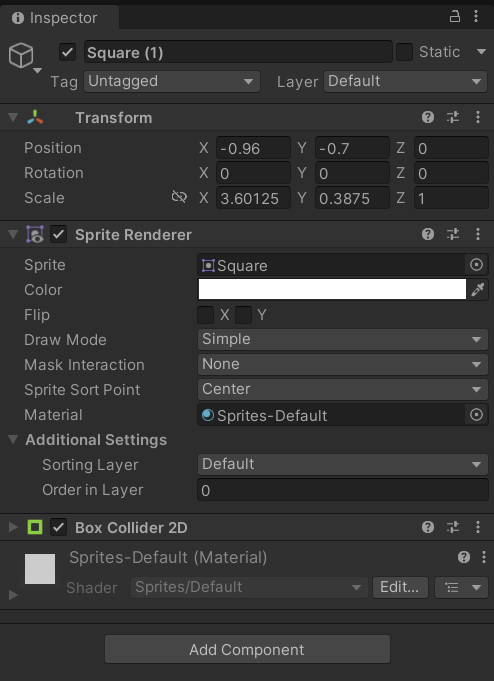
Hình 2.2. Giao diện hiển thị khung trò chơi

Hierarchy (Danh Sách Đối Tượng): Danh sách này hiển thị tất cả các đối tượng trong khung cảnh. Có thể tìm kiếm, sắp xếp và quản lý các đối tượng từ đây.



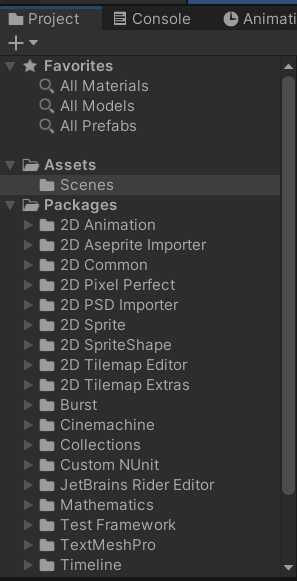
Hình 2.3. Giao diện hiển thị danh sách các đối tượng

Inspector (Bảng Thuộc Tính): Khi chọn một đối tượng trong danh sách, thông tin chi tiết về đối tượng đó sẽ hiển thị ở đây. Bạn có thể chỉnh sửa các thuộc tính của đối tượng từ bảng này. Là nơi mà ta có thể xem và chỉnh sửa các thuộc tính và thành phần của đối tượng. Tính năng này giúp các game designer cũng có thể tham gia vào quá trình làm game mà không cần biết quá nhiều về [code](https://cellphones.com.vn/sforum/tag/code-game).



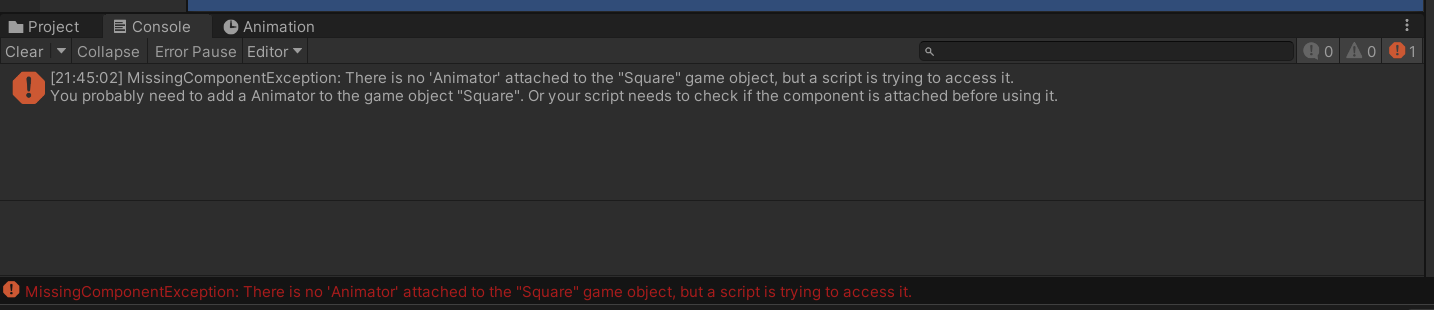
Hình 2.4. Giao diện bảng thuộc tính trong Unity

Project (Dự Án): Đây là nơi mà các developer quản lý tất cả tài nguyên, script và tệp liên quan đến dự án làm game. Developer có thể tổ chức các tệp theo thư mục để dễ dàng tìm kiếm và kéo - thả vào scene mình muốn.



Hình 2.5. Giao diện quản lý dự án trong Unity

Console: Bảng này hiển thị thông báo lỗi, cảnh báo và ghi chú từ Unity hoặc từ mã lập trình của bạn.



Hình 2.6. Giao diện thông báo lỗi trong Unity

Toolbar (Thanh Công Cụ): Thanh này chứa các công cụ để tạo, chỉnh sửa và quản lý trò chơi. Bạn có thể thay đổi chế độ xem, tạo đối tượng mới và thực hiện các tác vụ khác từ thanh này.



Hình 2.7. Giao diện thanh công cụ trong Unity

### 2.2.2. Các thành phần trong Unity Engine

* GameObject: Đây là khái niệm dùng để chỉ những đối tượng cụ thể trong game, nó có thể là bất kỳ một nhân vật hay đồ vật nào đó được xây dựng trong trò chơi.
* Component: Component là các phần mở rộng của GameObject, chứa dữ liệu và logic. Ví dụ, Transform Component xác định vị trí, quay và tỉ lệ của một GameObject.
* Scene: Là một thành phần không thể thiếu trong game, chứa các vật thể tương tác với thế giới, bao gồm cả Player của chúng ta.
* Prefab: Là khái niệm dùng để chỉ các đối tượng giống nhau có trong trò chơi, chỉ cần thực hiện thao tác khởi tạo lại các giá trị trong vị trí cùng các tỷ lệ biến dạng hay góc quay từ đối tượng ban đầu.
* Script: Thể hiện một tập tin chứa các đoạn mã nguồn, dùng để khởi tạo và xử lý các đối tượng trong game. Đối với Unity, bạn có thể sử dụng C#, Javascript, BOO để tạo ra các Script phục vụ cho công việc.
* Physics: Physics là hệ thống xử lý vật lý trong Unity, giúp xử lý va chạm, chuyển động và tương tác với vật thể.
* Material: Material xác định cách một đối tượng được hiển thị. Nó chứa thông tin về màu sắc, độ bóng, texture, và các thuộc tính đồ họa khác.
* Shader: Shader là một đoạn mã được sử dụng để xác định cách mà vật liệu phản ánh ánh sáng và màu sắc. Nó có thể tùy chỉnh đồ họa của vật liệu.
* Camera: Camera quyết định cách trò chơi được hiển thị cho người chơi. Các thành phần như cắt nhau, độ sâu trường và góc nhìn được xác định bởi camera.
* Light: Light xác định cách ánh sáng tương tác với vật thể trong Scene. Unity hỗ trợ nhiều loại ánh sáng như điểm sáng, ánh sáng hình học và ánh sáng khu vực.
* Rigidbody: Rigidbody được sử dụng để thêm vật lý cứng cho đối tượng. Điều này làm cho đối tượng phản ứng với lực, nặng, và các yếu tố vật lý khác.
* Collider: Collider xác định khu vực va chạm của đối tượng, giúp hệ thống vật lý xác định khi nào hai đối tượng chạm vào nhau và cách chúng phản ứng.

2.3.