

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  
**KHOA ĐA PHƯƠNG TIỆN**



**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.

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 bạn đọc sẽ có một trải nghiệm thú vị khi khám phá trò chơi “Run or Die” thông qua bài báo cáo của nhóm.

# MỤC LỤC

LỜI MỞ ĐẦU .....	i
LỜI CẢM ƠN.....	v
KÝ HIỆU VÀ VIẾT TẮT .....	vi
DANH MỤC CÁC HÌNH VẼ .....	x
DANH MỤC CÁC BẢNG .....	xii
Chương 1. CƠ SỞ LÝ THUYẾT VÀ TỔNG QUAN VỀ GAME .....	1
1.1. Tổng quan về game .....	1
1.1.1. Giới thiệu về Video game .....	1
1.1.2. Lịch sử phát triển game .....	1
1.1.3. Các thể loại game.....	2
1.2. Kiến trúc game .....	3
1.2.1. Kiến trúc game.....	3
1.2.2. Mô hình 3 lớp MVC .....	7
1.2.3. Mô hình MVVM.....	9
1.3. Quy trình phát triển game .....	10
1.3.1. Giai đoạn tiền sản xuất .....	10
1.3.2. Giai đoạn sản xuất .....	13
1.3.3. Giai đoạn hậu sản xuất.....	14
1.4. Phương pháp tiếp cận hướng đối tượng .....	16
1.4.1. Giới thiệu về phương pháp hướng đối tượng .....	16
1.4.2. Đặc trưng của phương pháp hướng đối tượng.....	16
1.4.3. Lợi ích và hạn chế của phương pháp hướng đối tượng .....	16
1.5. Tiêu kết chương 1 .....	17
Chương 2. NGHIÊN CỨU VỀ ỨNG DỤNG TRONG GAME.....	18
2.1. Unity Engine .....	18

---

2.1.1. Giới thiệu về Unity .....	18
2.1.2. Lịch sử hình thành và phát triển .....	18
2.2. Các giao diện chính và các thành phần trong Unity Engine .....	18
2.2.1. Các giao diện chính trong Unity .....	18
2.2.2. Các thành phần trong Unity Engine.....	22
2.3. Luồng xử lý trong Unity Engine .....	22
2.4. Tiểu kết chương 2 .....	23
Chương 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	24
3.1. Xác định yêu cầu bài toán .....	24
3.1.1. Yêu cầu bài toán .....	24
3.1.2. Đối tượng và phạm vi nghiên cứu .....	24
3.1.3. Mục tiêu và nhiệm vụ nghiên cứu .....	25
3.1.4. Tính cấp thiết của sản phẩm .....	25
3.2. Tổng quan về game “Run or Die” .....	26
3.2.1. Tổng quan về game – GDD One Page .....	26
3.2.2. Full GDD .....	27
3.3. Technical Design Document .....	44
3.3.1. Danh sách các tính năng dựa trên GDD .....	44
3.3.2. Các công cụ sử dụng.....	44
3.3.3. Bảng lịch trình .....	45
3.3.4. Quản lý scene.....	45
3.3.5. Nền tảng, yêu cầu phần cứng và phần mềm .....	46
3.4. Phân tích và thiết kế hệ thống .....	47
3.4.1. Biểu đồ Usecase.....	47
3.4.2. Kịch bản (Scenario) .....	47
3.4.3. Biểu đồ tuần tự (Sequence).....	50
3.4.4. Biểu đồ lớp thiết kế.....	54
3.5. Tiểu kết chương 3 .....	54

---

---

Chương 4. CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG .....	56
4.1. Cài đặt, triển khai hệ thống .....	56
4.1.1. Cài đặt công cụ Unity .....	56
4.1.2. Cài đặt IDE Visual Studio 2022 .....	58
4.1.3. Thiết kế wireframe.....	59
4.1.4. Một số màn hình chính trong game .....	60
4.1.5. Thiết kế màn chơi trong game .....	62
4.2. Đánh giá kết quả.....	63
4.3. Tiểu kết chương 4 .....	65
KẾT LUẬN .....	66
PHỤ LỤC .....	67
TÀI LIỆU THAM KHẢO .....	72

## 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 Die” 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

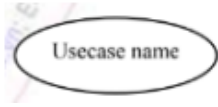

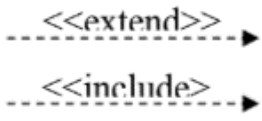

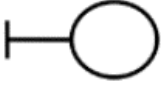

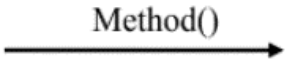
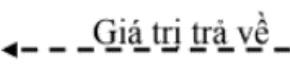
## KÝ HIỆU VÀ VIẾT TẮT

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.

Ký hiệu trong biểu đồ	Phần tử mô hình	Ý nghĩa
	Usecase	Biểu diễn một chức năng xác định của hệ thống
	Tác nhân	Là một đối tượng ngoài hệ thống tác động đến các usecase
	Mối quan hệ giữa các usecase	Tùy từng dạng quan hệ
	Lớp thực thể	
	Lớp biên (Lớp giao diện)	
	Lớp điều khiển	
	Gọi (call)	Mô tả lời gọi từ một đối tượng này tới đối tượng kia
	Trả về (return)	Trả về giá trị ứng với lời gọi

## DANH MỤC CÁC HÌNH VẼ

Hình 1.1. Game application layer.....	4
Hình 1.2. Game logic layer.....	5
Hình 1.3. Game view – human.....	6
Hình 1.4. Game view – AI.....	7
Hình 1.5. Mô hình 3 lớp MVC.....	8
Hình 1.6. Mô hình MVVM.....	9
Hình 1.7. Quy trình phát triển trò chơi.....	10
Hình 1.8. Tài liệu thiết kế trò chơi.....	12
Hình 1.9. Lập trình tích hợp.....	13
Hình 1.10. Kiểm thử và bảo trì.....	15
Hình 2.1. Giao diện hiển thị khung cảnh trong Unity.....	18
Hình 2.2. Giao diện hiển thị khung trò chơi.....	19
Hình 2.3. Giao diện hiển thị danh sách các đối tượng.....	19
Hình 2.4. Giao diện bảng thuộc tính trong Unity.....	20
Hình 2.5. Giao diện quản lý dự án trong Unity.....	21
Hình 2.6. Giao diện thông báo lỗi trong Unity.....	21
Hình 2.7. Giao diện thanh công cụ trong Unity.....	21
Hình 3.1. Mô tả các phím trong trò chơi.....	29
Hình 3.2. Giao diện trò chơi tham khảo.....	30
Hình 3.3. Màn hình tham khảo khi vào trò chơi.....	30
Hình 3.4. Màn hình tham khảo lựa chọn các màn chơi.....	31
Hình 3.5. Game flowchart.....	31
Hình 3.6. Minh họa hiển thị thanh máu của người chơi.....	32
Hình 3.7. Nhân vật Michael.....	32
Hình 3.8. Vật phẩm súng ACR.....	33
Hình 3.9. Vật phẩm bom.....	34
Hình 3.10. Mô tả đường bay đạn của súng ACR.....	34
Hình 3.11. Tham khảo màn hình thành tích người chơi.....	35
Hình 3.12. Biểu đồ nhịp.....	37

Hình 3.13. Địa hình trong game .....	41
Hình 3.14. Môi trường trong game.....	42
Hình 3.15. Vật phẩm thùng gỗ nhỏ .....	42
Hình 3.16. Vật phẩm thùng thuốc nổ .....	42
Hình 3.17. Vật phẩm thùng gỗ lớn .....	43
Hình 3.18. Bảng lịch trình triển khai kế hoạch phát triển game.....	45
Hình 3.19. Build Settings .....	45
Hình 3.20. Quản lý các scene .....	46
Hình 3.21. Biểu đồ Usecase tổng quát .....	47
Hình 3.22. Biểu đồ tuần tự chức năng start game .....	50
Hình 3.23. Biểu đồ tuần tự chức năng choose level.....	51
Hình 3.24. Biểu đồ tuần tự chức năng di chuyển của nhân vật.....	51
Hình 3.25. Biểu đồ tuần tự chức năng hiển thị giao diện setting .....	52
Hình 3.26. Biểu đồ tuần tự chức năng setting .....	53
Hình 3.27. Biểu đồ lớp thiết kế .....	54
Hình 4.1. Màn hình download Unity.....	56
Hình 4.2. Lựa chọn giấy phép sử dụng.....	57
Hình 4.3. Giao diện Unity sau khi cài đặt thành công.....	57
Hình 4.4. Giao diện cài đặt Visual Studio .....	58
Hình 4.5. Lựa chọn game development với unity .....	58
Hình 4.6. Thiết kế wireframe giao diện homescreen .....	59
Hình 4.7. Thiết kế wireframe giao diện levelselect.....	59
Hình 4.8. Thiết kế wireframe giao diện gameplay .....	60
Hình 4.9. Thiết kế wireframe giao diện game over.....	60
Hình 4.10. Giao diện home screen .....	61
Hình 4.11. Giao diện level select.....	61
Hình 4.12. Giao diện game play .....	62
Hình 4.13. Giao diện game over.....	62

---

## DANH MỤC CÁC BẢNG

Bảng 3.1. Danh sách các vật phẩm trong game.....	33
Bảng 3.2. Danh sách các nhân vật trong game.....	36
Bảng 3.3. Mô tả các level trong game .....	38
Bảng 3.4. Bảng mô tả quy tắc quái trong game .....	39
Bảng 3.5. Mô tả các boss trong game.....	41
Bảng 3.6. Đặc tả use case start game.....	48
Bảng 3.7. Đặc tả use case choose level .....	48
Bảng 3.8. Đặc tả use case setting .....	49
Bảng 3.9. Đặc tả use case bật tắt sound và music .....	49
Bảng 3.10. Đặc tả use case character controller.....	50
Bảng 4.1. Đánh giá kết quả .....	61

## **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 arcade 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 game 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

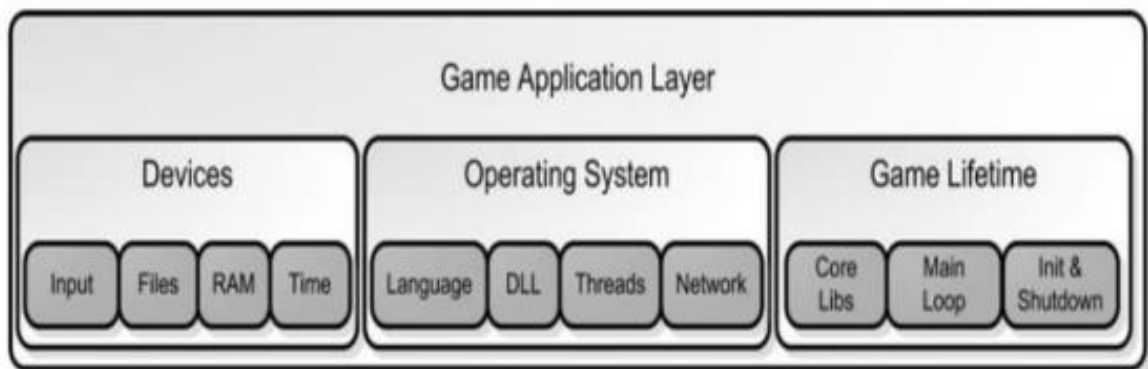
Kiến trúc game là cách mà các thành phần khác nhau trong một trò chơi máy tính được tổ chức và tương tác với nhau. Nó bao gồm cả cấu trúc tổ chức và thiết kế của trò chơi, cũng như các yếu tố kỹ thuật và công nghệ sử dụng để triển khai trò chơi đó.

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,...



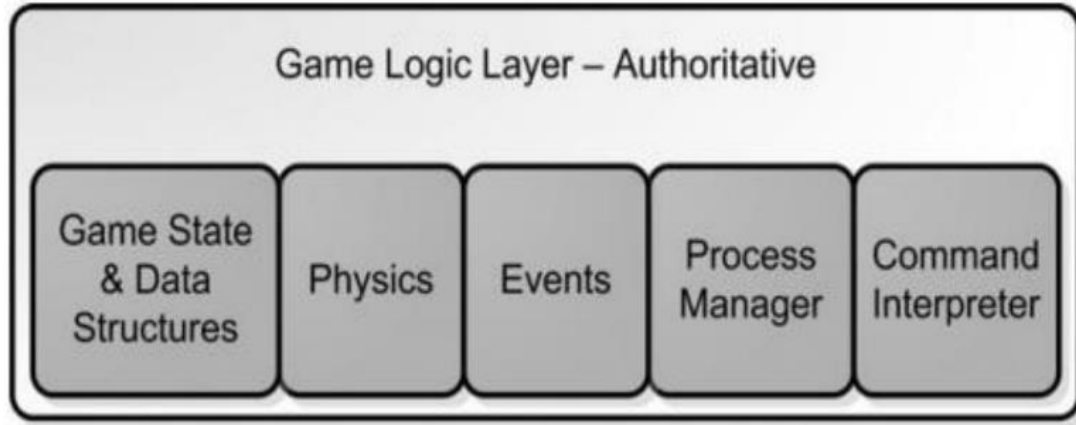
- + Quản lý thông tin trao đổi giữa game với hệ thống file: DVD/CD-ROM, đĩa cứng, thẻ nhớ.
- + Quản lý thông tin trao đổi giữa game với bộ nhớ đệm tài nguyên: Tài nguyên ko nên được tải trực tiếp từ nguồn Assets trong Project mà cần bộ đệm để tải, lưu trữ trước những tài nguyên đó → Đảm bảo tài nguyên 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 thể giới quan của game: trạng thái của game, những đối tượng có trong game và tương tác giữa chúng với nhau.

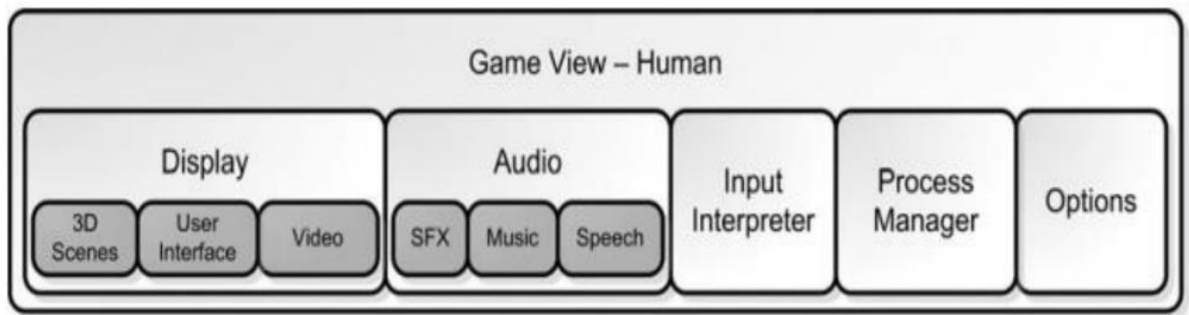


Hình 1.2. Game logic layer

- + 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:

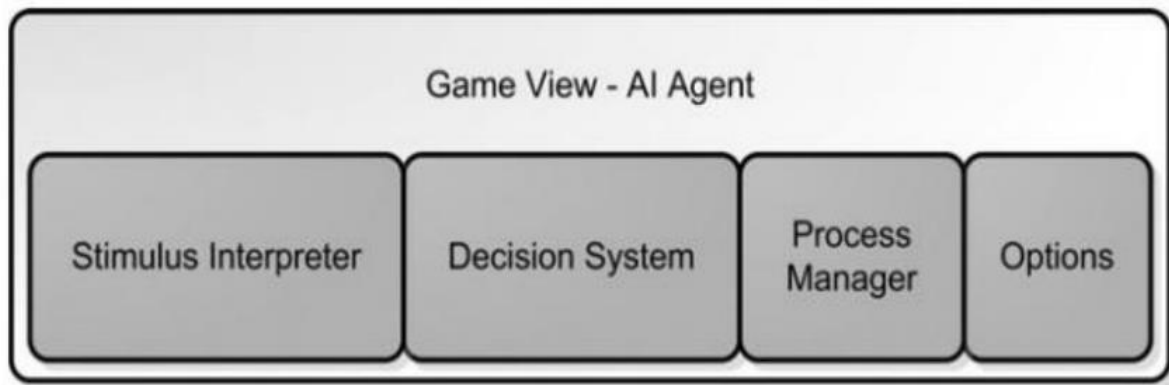
- Giao diện game là một tập hợp hệ thống tương tác với game logic để hiển thị giao diện game với từng vai game .
- Vai game: người chơi / đối tượng AI.



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.

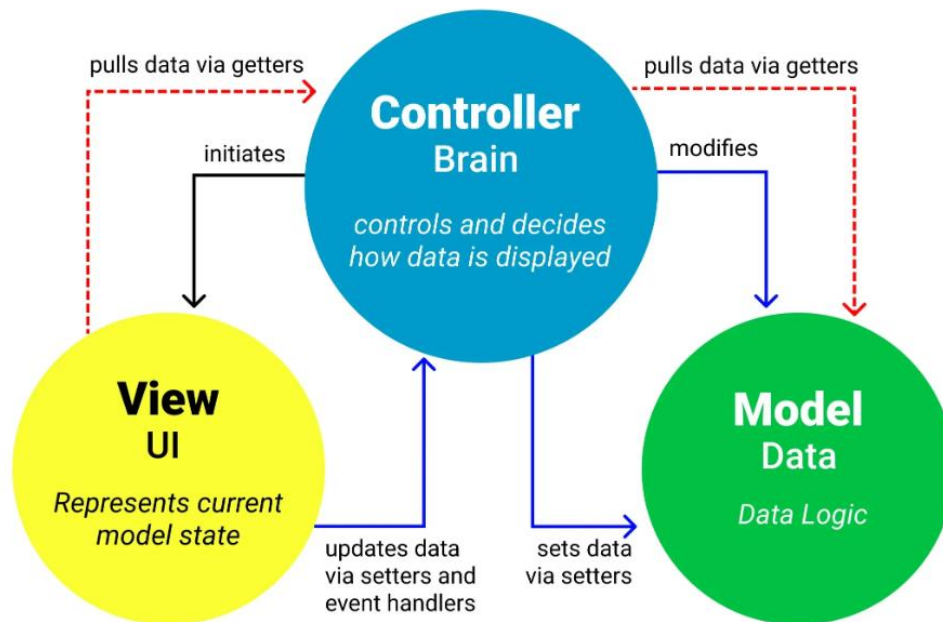


Hình 1.4. Game view – AI

### 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.

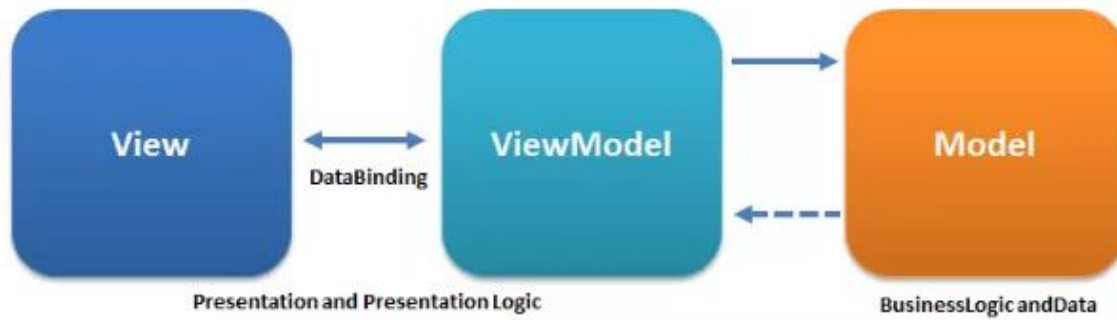
# MVC Architecture Pattern



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

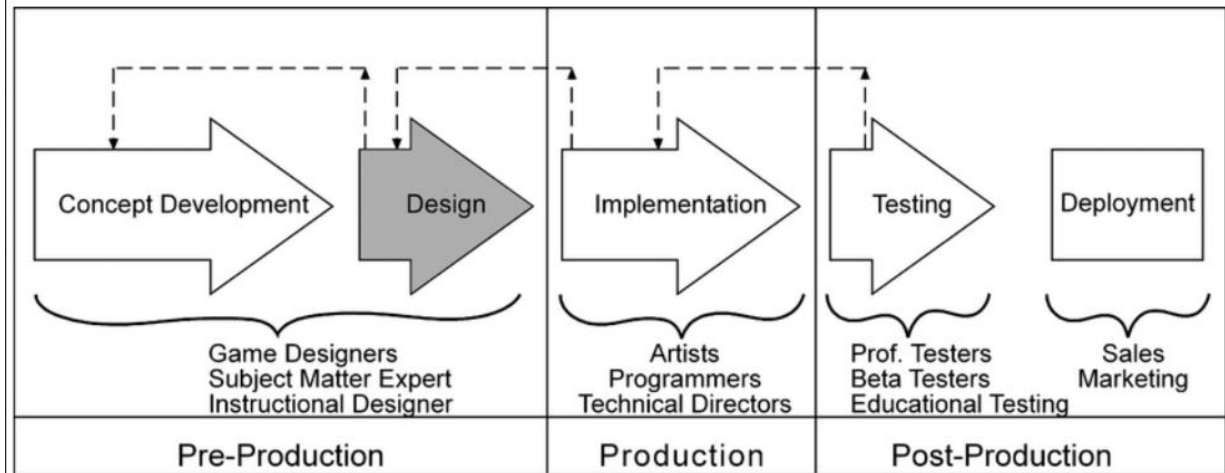


Hình 1.6. Mô hình MVVM

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.



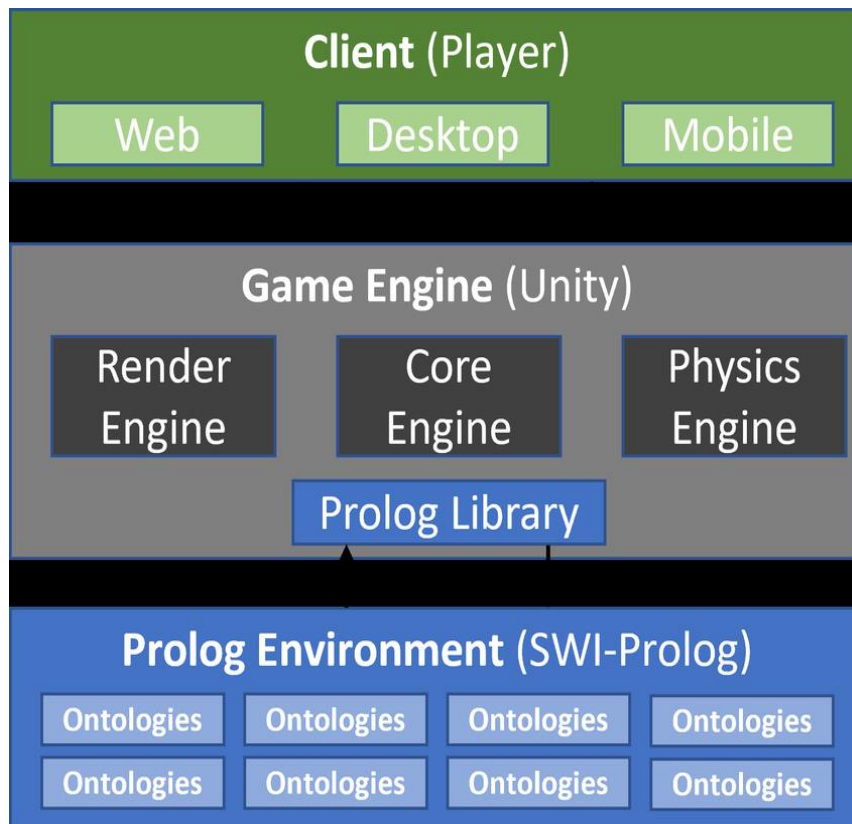
Hình 1.8. Tài liệu thiết kế trò chơi

- 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 triển game trên PC, consoles, thiết bị di động và trên websites.

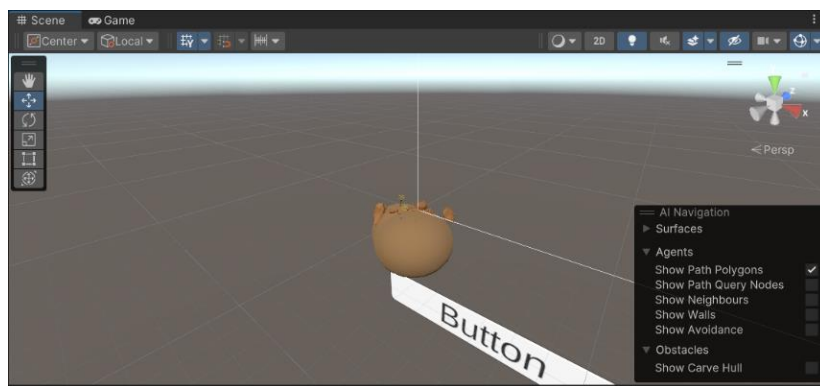
#### 2.1.2. Lịch sử hình thành và phát triển

Unity là một game engine đa nền tảng 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 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

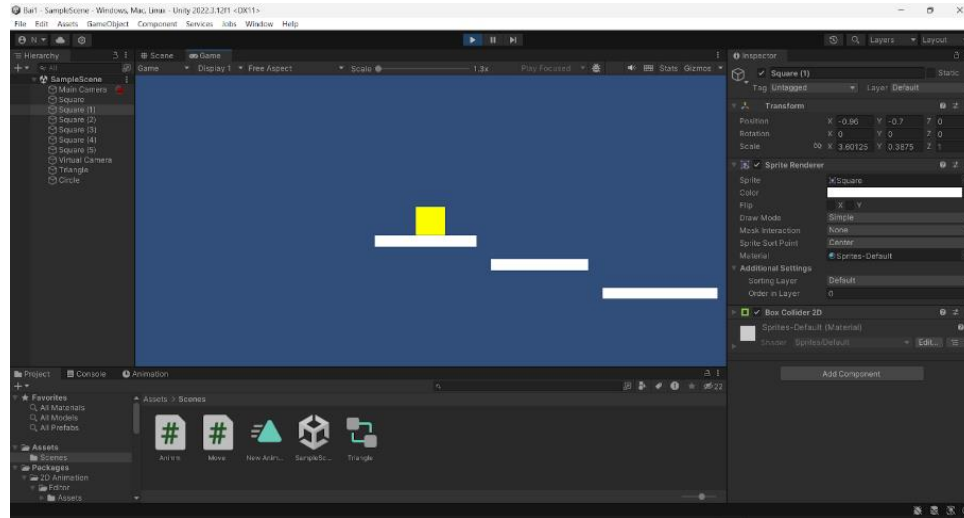
#### 2.2.1. Các giao diện chính trong Unity

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 camera, 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.



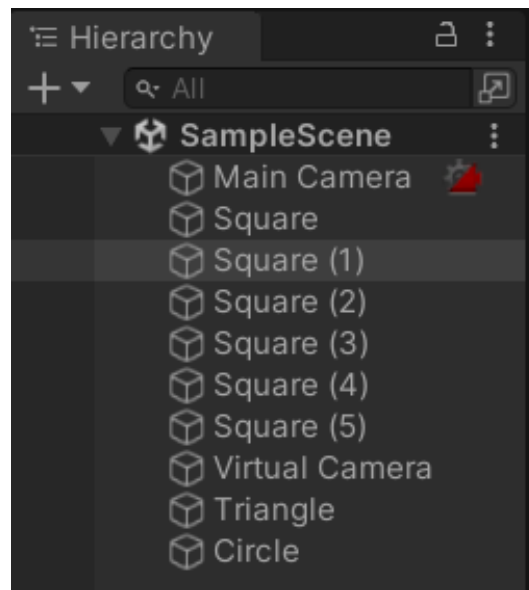
Hình 2.1. Giao diện hiển thị khung cảnh trong Unity

**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.

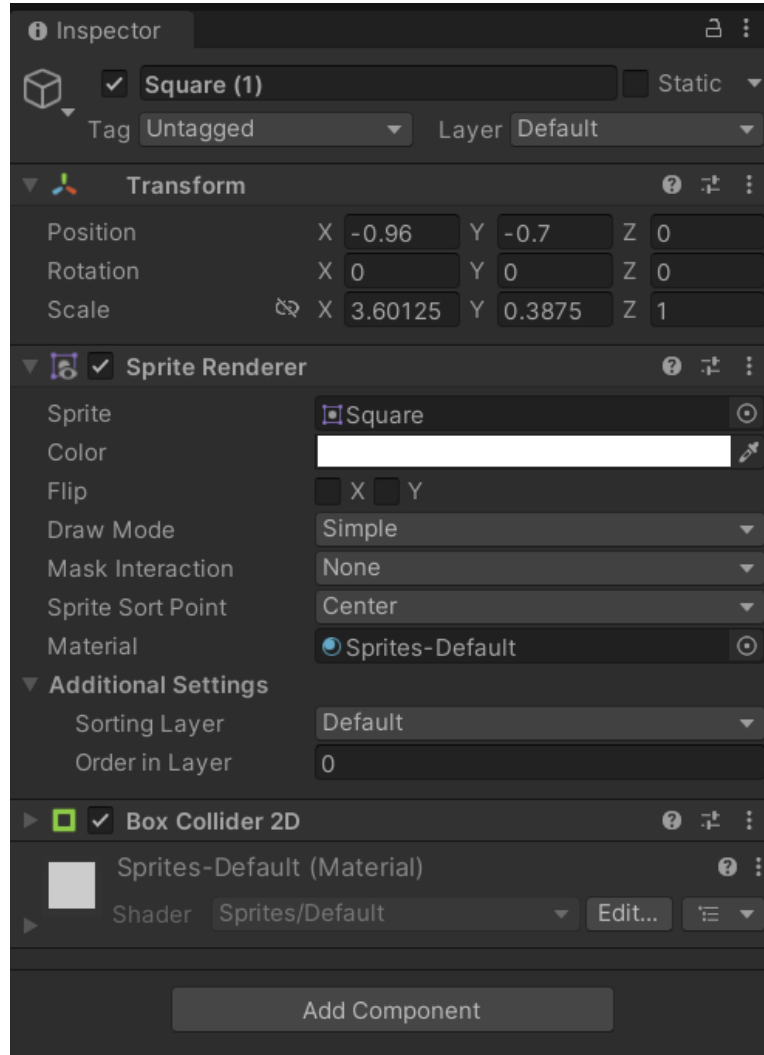


Hình 2.3. Giao diện hiển thị danh sách các đối tượng

**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

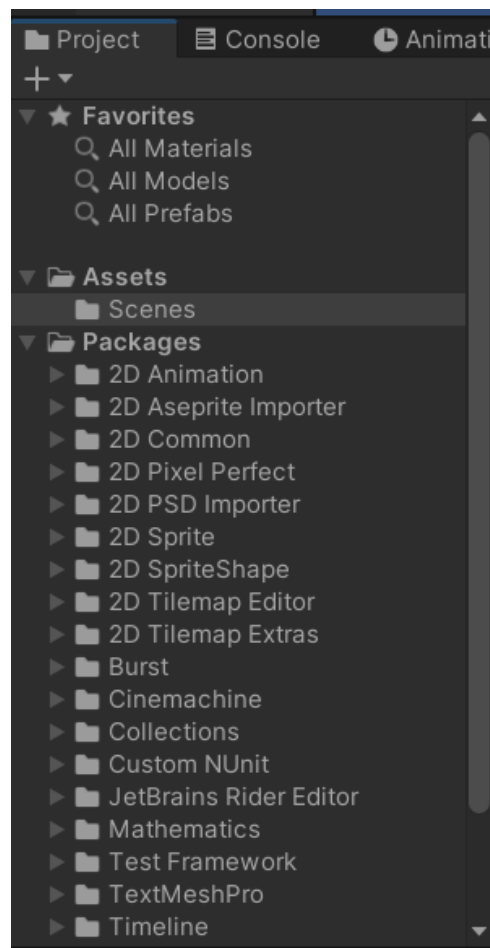


tượng. Tính năng này giúp các game designer cũng có thể tham gia vào quá trình làm game mà không cần biết quá nhiều về code.



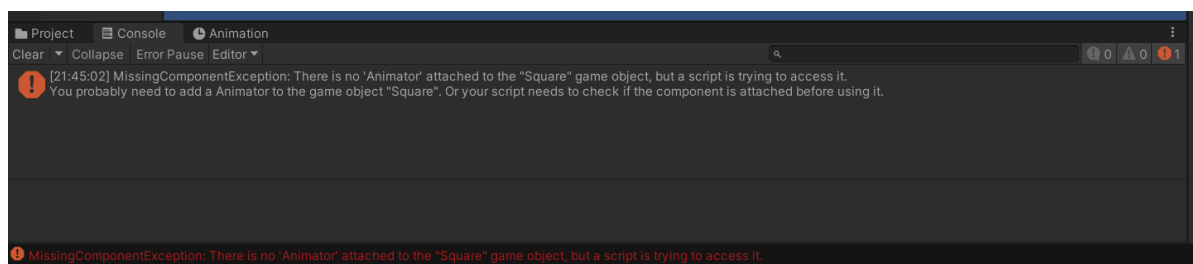
Hình 2.4. Giao diện bảng thuộc tính trong Unity

**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.



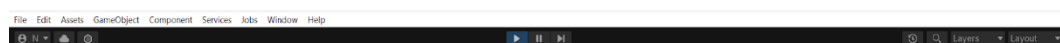
Hình 2.5. Giao diện quản lý dự án trong Unity

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. Luồng xử lý trong Unity Engine

Trong Unity, “luồng xử lý” thường được hiểu là luồng chính của trò chơi, nơi mà toàn bộ logic của trò chơi diễn ra. Unity chủ yếu chạy trò chơi trong một luồng duy nhất, thường được gọi là “luồng chính” (main thread). Điều này có nghĩa là tất cả các hoạt động

chính như cập nhật (update), vẽ (rendering), và xử lý sự kiện người chơi đều xảy ra trong một luồng duy nhất. Trong Unity, có một số hàm quan trọng trong vòng đời của đối tượng (GameObject) hoặc một kịch bản (Script). Ba hàm quan trọng nhất trong số này là “Awake()”, “Start()”, “Update()”:

- Awake():
  - + Mô tả: Hàm này được gọi khi một đối tượng được tạo ra kích thích.
  - + Sử dụng: Thường được sử dụng để khởi tạo các tham chiếu đối tượng, thực hiện các công việc chuẩn bị trước khi trò chơi bắt đầu.
- Start():
  - + Mô tả: Hàm “Start()” được gọi sau khi tất cả các hàm “Awake()” đã được gọi.
  - + Sử dụng: Thường được sử dụng để thiết lập các giá trị khởi đầu, tạo các thực thể, và thực hiện các công việc cần thực hiện khi trò chơi bắt đầu.
- Update():
  - + Mô tả: Hàm “Update()” được gọi sau mỗi frame
  - + Sử dụng: Thường được sử dụng để cập nhật trạng thái của đối tượng, xử lý đầu vào từ người chơi, và thực hiện các tính toán cần thiết trong mỗi frame.

## 2.4. Tiểu kết chương 2

Chương 2 đã mở đầu bằng một cái nhìn tổng quan về Unity Engine, cung cấp cho chúng ta một cơ sở vững chắc về lịch sử, giao diện ứng dụng cơ bản, và các khái niệm quan trọng. Đây thực sự là một tiền đề quan trọng để chuẩn bị cho việc khám phá sâu hơn về phát triển trò chơi và ứng dụng trên nền tảng mạnh mẽ này.

Trong chương tiếp theo, chúng em sẽ dành thời gian để tập trung vào phân tích và thiết kế hệ thống của trò chơi. Chúng em sẽ tổng quan về cấu trúc cơ bản của trò chơi, tìm hiểu về các yếu tố quan trọng như gameplay, đồ họa, âm thanh, và tương tác người chơi. Việc này sẽ đặt nền móng cho quá trình phát triển và giúp chúng ta hiểu rõ hơn về cách Unity Engine có thể được tận dụng để tạo ra những trải nghiệm chơi game độc đáo.

Ngoài ra, chúng em sẽ đề cập đến các tài liệu thiết kế game (GDD), một công cụ quan trọng để tạo ra hướng dẫn chi tiết về cấu trúc và chức năng của trò chơi. Việc này sẽ giúp chúng ta duy trì sự nhất quán trong quá trình phát triển và tạo điều kiện cho sự hiểu biết chung giữa các thành viên trong nhóm làm việc.

## **Chương 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

### **3.1. Xác định yêu cầu bài toán**

#### **3.1.1. Yêu cầu bài toán**

- Game thuộc thể loại hành động - chạy đua → Cần đem lại trải nghiệm chơi thú vị cho người chơi qua việc thiết kế nhân vật, giao diện, thiết kế các màn chơi đa dạng.
- + Nhân vật trong trò chơi cần được thiết kế từ những chi tiết nhỏ như trang phục, biểu cảm khuôn mặt, và đặc điểm nổi bật của nhân vật có thể tạo nên sự liên kết giữa người chơi và nhân vật, thúc đẩy sự tương tác và hứng thú.
- + Giao diện trò chơi cần được thiết kế sao cho dễ sử dụng và hấp dẫn. Các nút điều khiển và thông tin trên màn hình phải được sắp xếp một cách logic, không gian trống phải được tận dụng hiệu quả để không làm mất tập trung của người chơi.
- + Việc thiết kế các màn chơi đa dạng là chìa khóa để giữ cho trò chơi luôn mới mẻ và thách thức. Sự đổi mới trong cảnh quay, thách thức tăng dần, và những yếu tố ngẫu nhiên có thể giúp tránh tình trạng lặp lại. Ngoài ra, cần chú ý tới độ khó của mỗi màn chơi để đảm bảo sự cân bằng giữa thách thức và niềm vui của người chơi.
- Đồ họa và âm nhạc trong game phải hấp dẫn → Tạo trải nghiệm thú vị và sôi động cho người chơi. Âm nhạc năng động và hiệu ứng âm thanh sinh động có thể kích thích cảm giác hứng thú và giữ cho người chơi trải nghiệm từ đầu tới cuối.

#### **3.1.2. Đối tượng và phạm vi nghiên cứu**

- Đối tượng nghiên cứu: Nghiên cứu được đặt mục tiêu chủ yếu vào nhóm người chơi đam mê thể loại game 2D hành động – chạy đua. Đối tượng nghiên cứu chính là các trẻ nam từ 15 tuổi trở lên. Đây là độ tuổi và giới tính mà trò chơi hướng đến, thông qua việc nghiên cứu đối tượng này, ta có thể đặt ra những chiến lược thiết kế và cải tiến trò chơi phù hợp với mong muốn của đối tượng người chơi chính.
- Phạm vi nghiên cứu: Nghiên cứu về gameplay và trải nghiệm người chơi trong thể giới 2D của trò chơi.
  - + Nghiên cứu sẽ khảo sát sự hấp dẫn và đa dạng của thể giới 2D trong trò chơi, bao gồm cả các yếu tố như đồ họa, môi trường và bối cảnh.
  - + Điều tra cảm nhận của người chơi về đồ họa và âm nhạc, đặc biệt là những yếu tố nào góp phần làm tăng trải nghiệm chơi game của họ.

- + Thu thập ý kiến, phản hồi từ đối tượng nghiên cứu về những cải tiến mong muốn và đề xuất giải pháp để tối ưu hóa trải nghiệm chơi game.

### 3.1.3. Mục tiêu và nhiệm vụ nghiên cứu

- Mục tiêu nghiên cứu: Tạo ra 1 trải nghiệm chơi game độc đáo và thu hút. Trong quá trình thiết kế và phát triển, mục tiêu là tạo ra một trò chơi mà người chơi không chỉ cảm nhận được sự hứng thú ngay từ đầu mà còn bị cuốn hút và giữ chân qua từng cấp độ.
- Nhiệm vụ nghiên cứu: Tập trung thiết kế các cấp độ với các nhiệm vụ và thách thức đa dạng để đem lại sự phấn khích và hứng thú cho người chơi.
  - + Nhiệm vụ và thách thức: Tập trung thiết kế các cấp độ với nhiệm vụ và thách thức đa dạng. Mỗi cấp độ cần mang lại trải nghiệm mới mẻ và thú vị, kết hợp giữa yếu tố hấp dẫn và độ khó phù hợp để giữ cho người chơi luôn được thách thức.
  - + Sự phấn khích và hứng thú: Tạo ra các yếu tố trong game như sự phấn khích và hứng thú thông qua cấu trúc cấp độ, việc đặt ra các nhiệm vụ cụ thể và các điểm nhấn hấp dẫn. Sự đổi mới và sự sáng tạo cần được tích hợp để tạo ra trải nghiệm chơi game không ngừng nghỉ.
  - + Đảm bảo có sự liên kết và mạch lạc giữa các cấp độ để người chơi cảm thấy như mình đang tham gia vào một cuộc phiêu lưu liên tục. Sự chuyển động từ một cấp độ sang cấp độ khác cần được thiết kế một cách mượt mà và logic.
  - + Phát triển các yếu tố tương tác trong game để người chơi có cơ hội tham gia một cách linh hoạt và thú vị. Các tương tác này có thể bao gồm cách nhân vật di chuyển, sử dụng vật phẩm, tương tác với môi trường.

### 3.1.4. Tính cấp thiết của sản phẩm

- Giải trí và giảm stress: Sản phẩm mang lại tính giải trí và giúp giảm căng thẳng cho người chơi trong những khoảnh khắc giải trí.
- + Một trong những mục tiêu hàng đầu của “Run or Die” là tạo ra một trải nghiệm giải trí tuyệt vời cho người chơi. Từ cấp độ đơn giản những đến những thách thức cao cấp, trò chơi mang lại niềm vui và sự hứng khởi trong mỗi khoảnh khắc chơi.

- + Trong những thời điểm căng thẳng, “Run or Die” không chỉ là một trò chơi, mà là một phương tiện giúp giảm stress. Việc tham gia vào thế giới 2D hấp dẫn này không chỉ mang lại niềm vui, mà còn giúp người chơi xả stress và tìm lại sự thư giãn.
- Khám phá và sáng tạo: Tính cấp thiết nằm ở khả năng khám phá và sáng tạo, mang đến một thế giới 2D mới mẻ và đầy thú vị.
  - + “Run or Die” đặt sự cấp thiết vào khả năng khám phá. Mỗi cấp độ và màn chơi được thiết kế để khám phá, giúp người chơi cảm thấy như họ đang mở ra một thế giới mới đầy bí ẩn và thú vị.
  - + Thế giới 2D của trò chơi không chỉ là nơi để chạy đua, mà còn là một nền tảng cho sự sáng tạo. Tính cách điệu của từng cấp độ, yếu tố tương tác, và các phần khác nhau của game đều được thiết kế để kích thích sự sáng tạo của người chơi.

### **3.2. Tổng quan về game “Run or Die”**

#### **3.2.1. Tổng quan về game – GDD One Page**

- Tên game: Run or Die
- Thể loại: 2D, hành động - phiêu lưu, chạy đua.
- Nền tảng phát hành: DESKTOP
- Đối tượng người chơi:
  - + Độ tuổi hướng tới: Nam từ 15-21 tuổi.
  - + Phù hợp với đối tượng trẻ em hứng thú, có sở thích với thể loại game phiêu lưu - chạy đua, đam mê khám phá.
- Các sản phẩm tương tự: Mario, Chicken invader, ...
- Game summary (Tóm tắt trò chơi): Michael là một người nông dân chuyên đi lấy củi trên núi. Một hôm nọ khi đang đi trên đường, một thực thể kỳ lạ xuất hiện và dùng phép dịch chuyển Michael tới một chiều không gian khác. Michael bị đưa tới 1 thế giới khác, thế giới này giống hệt thế giới nơi anh sống nhưng tuy nhiên bầu trời luôn là buổi hoàng hôn. Michael không chỉ là người duy nhất ở thế giới này mà còn có những quái vật khát máu và điên cuồng. Michael vừa phải chạy trốn khỏi lũ quái vật vừa phải tìm lối thoát khỏi thế giới kỳ quái này.
- Game Outline (Sơ lược trò chơi): Người chơi điều khiển nhân vật Michael, khéo léo tránh lũ quái vật và đạn của chúng trong một không gian cố định. Ngoài ra, người

chơi có thể điều khiển nhân vật bắn đạn để tiêu diệt quái vật. Trải qua các màn chơi, người chơi nhận được tiền và các vũ khí để nâng cấp sức mạnh cho nhân vật. Số lượng quái vật và các chướng ngại vật cũng sẽ gia tăng qua các màn chơi.

- USP:

- + Nhân vật: Người chơi có quyền thay đổi trang phục, vũ khí của nhân vật, nâng cấp nhân vật. Điều này đem lại sự mới mẻ cho cá tính của nhân vật.
- + Gameplay: Đa dạng, game cung cấp các màn chơi khác nhau với các quái vật khác nhau, các chướng ngại vật khác mà nhân vật có thể gặp phải trên đường chạy, các môi trường khác nhau từ sa mạc, đồi núi, biển cả..... Người chơi sẽ không bị nhàm chán khi chỉ chạy duy nhất trên 1 cung đường.
- + Đồ họa: Game có đồ họa 2D, hệ thống nâng cấp mở khóa, cho phép nhân vật được nâng cấp bằng những vật phẩm kiếm trong game, cải thiện khả năng chơi và thành tích tốt hơn.
- + Trải nghiệm người dùng: Game có giao diện đẹp mắt, điều hướng dễ dàng, nhanh chóng. Quá trình tải game không để người chơi chờ quá lâu, âm thanh phù hợp với không khí phiêu lưu của game.

### 3.2.2. Full GDD

- Tóm tắt trò chơi:

Lấy bối cảnh vùng núi nước Pháp trong những năm 80 của thế kỷ 20, Michael - một người nông dân thuộc vùng núi xa xôi, mỗi ngày anh ta gắng gượng lên núi để lấy củi và dự trữ cho gia đình. Cuộc sống yên bình của anh bị đảo lộn vào một buổi chiều tĩnh lặng. Khi anh đang băng băng trên con đường đá dốc, một thực thể kỳ lạ bất ngờ xuất hiện trước mặt anh. Trước khi Michael kịp tỏ ra bất kỳ sự sợ hãi nào, thực thể ấy dùng một phép dịch chuyển mạnh mẽ và cuốn anh vào một chiều không gian hoàn toàn khác biệt.

Michael bất ngờ rơi vào một thế giới đầy bí ẩn và kỳ quái. Dưới ánh hoàng hôn vĩnh cửu, bầu trời luôn trong trạng thái rực rỡ của buổi hoàng hôn. Nơi đây, thực tế có vẻ giống với thế giới mà anh đã biết, nhưng cảm giác khác biệt vẫn hiện hữu. Michael nhận ra mình không phải là người duy nhất ở thế giới này. Nơi này đầy rẫy những quái vật khát máu và điên cuồng, luôn rình rập trong bóng tối, chờ đợi cơ hội để tấn công.

May mắn, Michael không phải đối mặt với thế giới này một mình. Anh gặp gỡ các nhân vật đầy độc đáo và hỗ trợ trong cuộc hành trình đầy hiểm nguy. Trong số họ, có Hogwarts, một chiến binh mạnh mẽ, sẵn sàng đứng cùng Michael chống lại quái vật.



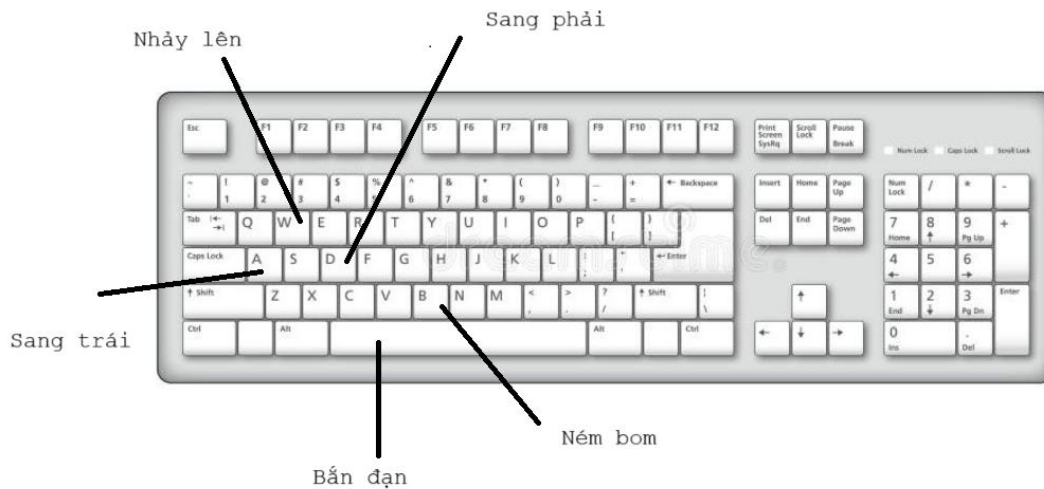
Timo, một sơn thần tài năng, mang theo những phép thuật bí ẩn. Jasmine, một người xạ thủ tài ba, có thể tạo ra vũ khí mạnh mẽ để tấn công bè lũ quái vật.

Michael bây giờ phải đối mặt với hai nhiệm vụ cực kỳ khó khăn: trốn thoát khỏi lũ quái vật đáng sợ này và tìm cách trở về thế giới của mình. Cuộc hành trình đầy thách thức và kỳ quái này đã bắt đầu, và Michael và những người đồng hành của anh sẽ phải đối mặt với nhiều điều bất ngờ, nguy hiểm và bí ẩn trước khi họ có thể tìm lối thoát khỏi thế giới kỳ lạ này. Liệu Michael có thể thoát khỏi lũ quái vật và trở về với thế giới của mình?

- GDD Outline:

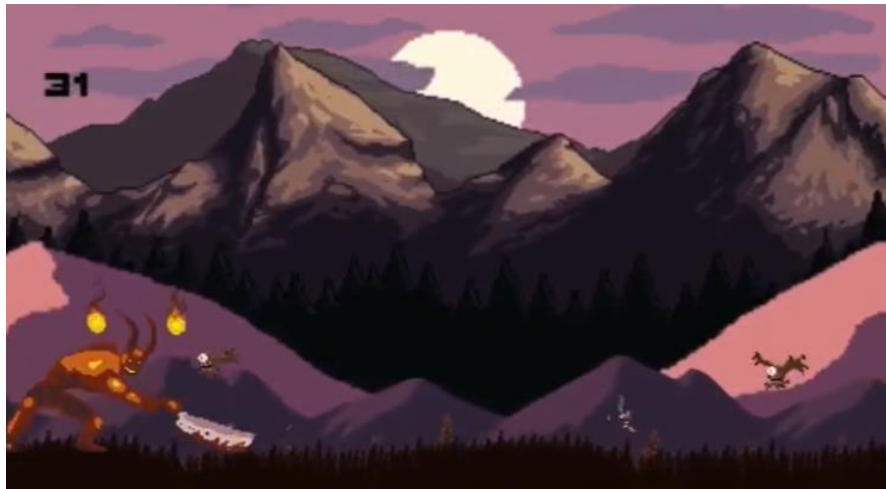
- + Lịch sử sửa đổi: Phiên bản đầu tiên - 1.0
- + Game goals: Trò chơi là một cuộc hành trình chạy đua để trốn thoát khỏi lũ quái vật và tìm kiếm lối thoát của nhân vật Michael.
- + Story overview:
  - Người chơi vào vai nhân vật Michael, thực hiện chọn vũ khí và trang phục, sau đó vượt qua các thử thách.
  - Người chơi điều khiển nhân vật Michael, khéo léo tránh lũ quái vật và đạn của chúng trong một không gian cố định. Ngoài ra, người chơi có thể điều khiển nhân vật bắn đạn để tiêu diệt quái vật.
  - Trải qua các màn chơi, người chơi nhận được tiền và các vũ khí để nâng cấp sức mạnh cho nhân vật. Số lượng quái vật và các chướng ngại vật cũng sẽ gia tăng qua các màn chơi.
  - Các địa điểm sẽ được sắp xếp theo trình tự diễn biến cốt truyện. Quá trình chơi game, nhân vật Michael sẽ gặp các nhân vật phụ khác và họ sẽ cùng đồng hành cùng Michael trong 1 số level nhất định (hỗ trợ chiến đấu).
  - Mỗi 5-7 level, người chơi được đưa tới 1 vùng đất khác phù hợp với sự phát triển nội dung câu chuyện.
- + Game controls:
  - Người chơi sử dụng các phím trên bàn phím để điều khiển nhân vật di chuyển hoặc tấn công kẻ thù.
    - W: Nhảy lên
    - A: Sang trái
    - D: Sang phải

- Space: Bắn đạn
- B: Ném bom



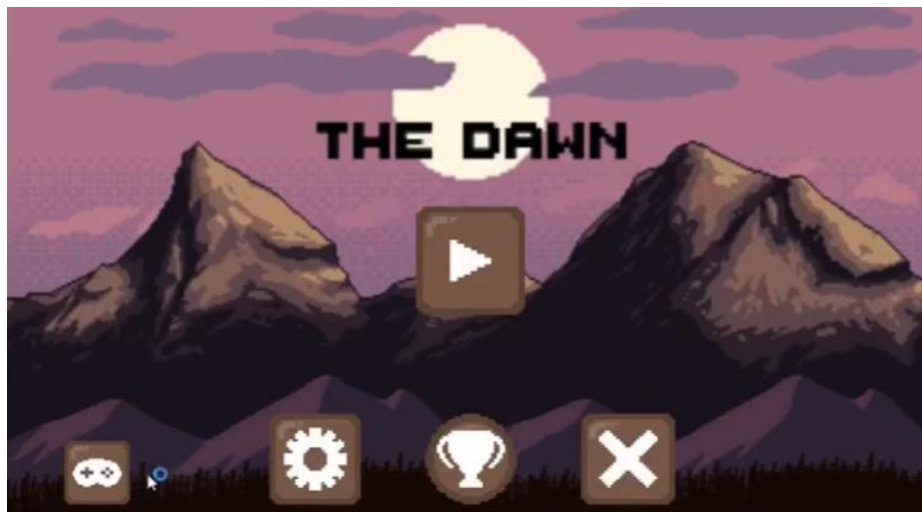
Hình 3.1. Mô tả các phím trong trò chơi

- Sử dụng chuột để lựa chọn các chức năng (chọn các màn chơi, thay đổi trang phục hay vũ khí, thiết lập các cài đặt khác).
- + Yêu cầu về công nghệ: Sử dụng trên nền tảng desktop, ngoài kết nối bàn phím và chuột, có thể sử dụng thêm tai nghe để tận hưởng trọn vẹn âm thanh và âm nhạc của game.
- + Trải nghiệm chơi game:
  - Nhân vật: Người chơi có quyền thay đổi trang phục, vũ khí của nhân vật, nâng cấp nhân vật. Điều này đem lại sự mới mẻ cho cá tính của nhân vật.
  - Gameplay: Đa dạng, game cung cấp các màn chơi khác nhau với các quái vật khác nhau, các chướng ngại vật khác mà nhân vật có thể gặp phải trên đường chạy, các môi trường khác nhau từ sa mạc, đồi núi, biển cả..... Người chơi sẽ không bị nhàm chán khi chỉ chạy duy nhất trên 1 cung đường.
  - Đồ họa: Game có đồ họa 2D, hệ thống nâng cấp mở khóa, cho phép nhân vật được nâng cấp bằng những vật phẩm kiếm trong game, cải thiện khả năng chơi và thành tích tốt hơn.
  - Trải nghiệm người dùng: Game có giao diện đẹp mắt, điều hướng dễ dàng, nhanh chóng. Quá trình tải game không để người chơi chờ quá lâu, âm thanh phù hợp với không khí phiêu lưu của game.
- + Giao diện trò chơi:



Hình 3.2. Giao diện trò chơi tham khảo

- + Cutscene description: Sau khi vào game, hiển thị cảnh chiến đấu trong game ngắn, sau đó chuyển vào màn hình bắt đầu game.
- + Title/start screen: Màn hình bắt đầu trò chơi với logo game đặt ở góc trên màn hình và 4 button với các tiêu đề play, highscore, setting, quit.



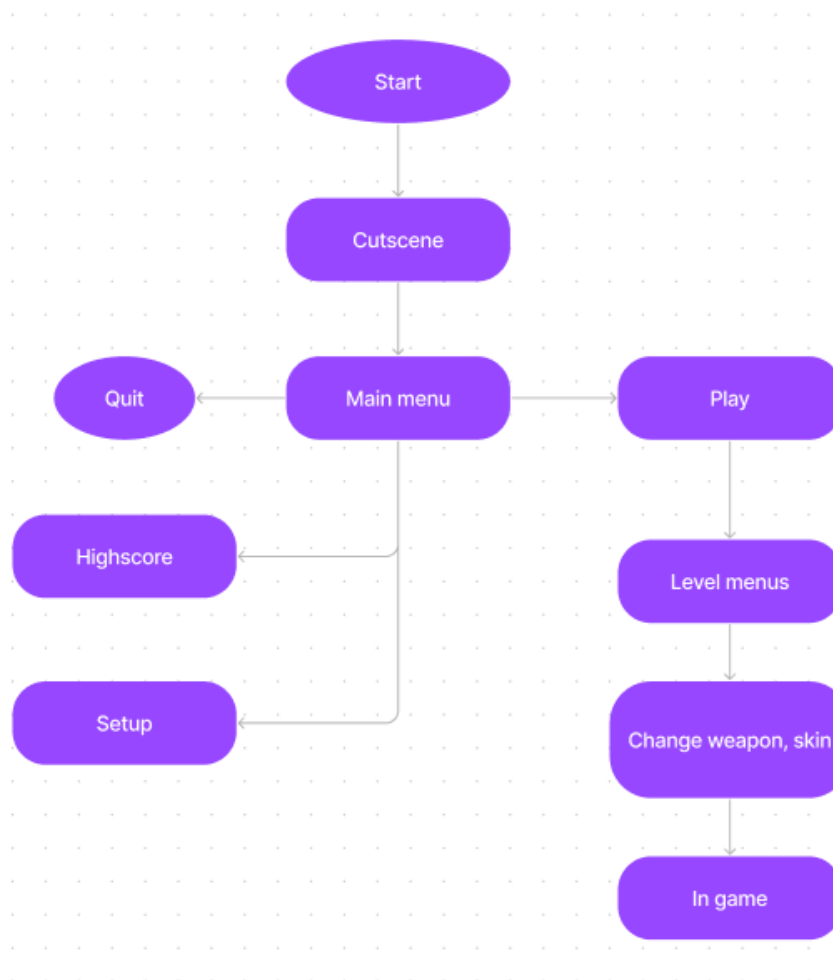
Hình 3.3. Màn hình tham khảo khi vào trò chơi

- + Level select / navigation screen: Sau khi click button play sẽ chuyển tới màn hình chọn các level để chơi.



Hình 3.4. Màn hình tham khảo lựa chọn các màn chơi

- + Other screens: Màn hình setting, highscore....
- + Game flowchart:



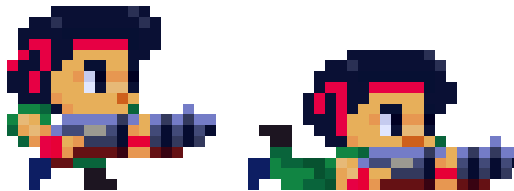
Hình 3.5. Game flowchart

- + Game camera(s):
  - Sử dụng góc nhìn ngang, chỉ thấy được bên trái hoặc bên phải của nhân vật, cảnh vật chỉ có 2D không xoay chuyển để nhìn thấy toàn bộ.
  - Các tình huống đặc biệt: Khi bắt đầu trò chơi, camera zoom out từ nhân vật, dần hiển thị không gian trong màn chơi.
- + HUD systems: Giao diện trong màn chơi với thanh máu hiển thị góc trái trên màn hình.




Hình 3.6. Minh họa hiển thị thanh máu của người chơi

- + Player character(s): Michael: Nhân vật người chơi tham gia vào cuộc hành trình chạy trốn khỏi lũ quái vật. Trong một vài thời điểm, nhân vật sẽ được hỗ trợ bởi một vài nhân vật khác, cụ thể đó là Hogwarts, Jasmine, Timo, Mario là những người đồng hành trong một số màn chơi nhất định.



Hình 3.7. Nhân vật Michael

- + Player inventory tools:
  - Danh sách vật phẩm bao gồm vàng, vật phẩm hồi máu, vật phẩm tăng tốc độ di chuyển

Vật phẩm	Tên vật phẩm	Mô tả
	Hồi máu	Lập tức hồi 4 đơn vị máu trong 2 giây
	Vàng	Cho 10 vàng, xuất hiện ngẫu nhiên trong các màn chơi
	Tăng tốc độ di chuyển	Tăng tốc độ di chuyển thêm 30% trong 10 giây

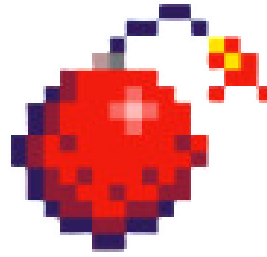
Bảng 3.1. Danh sách các vật phẩm trong game

- + Inventory Screen: Chế tạo lắp ghép các mảnh nhỏ thành đồ lớn.
- + Weapon progression:
  - Súng ACR: Súng thường đạn vô hạn, mỗi viên gây 3 đơn vị máu.

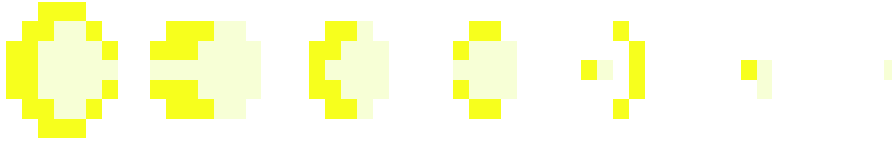


Hình 3.8. Vật phẩm súng ACR

- Súng Ak5: Dạng nâng cấp của súng thường gây 6 đơn vị máu mỗi lần bắn, nhưng chỉ có 100 viên đạn.
- Súng CZ: Súng bắn đạn lửa, gây 4 đơn vị máu và đốt 2 đơn vị máu mỗi giây trong vòng 5 giây.
- Súng QBZ: Súng có sát thương cao, có một mũi lê gắn ở đầu súng, bắn được 1 viên mỗi lần bắn, gây 20 đơn vị máu, gồm 20 viên đạn. Khi không bắn có thể dùng lưỡi lê tấn công cận chiến, gây sát thương 10 đơn vị máu.
- Bom: Ném bom gây sát thương, mỗi vụ nổ gây sát thương diện rộng và 12 đơn vị máu. Gồm 5 quả bom, hết không thể sạc lại.

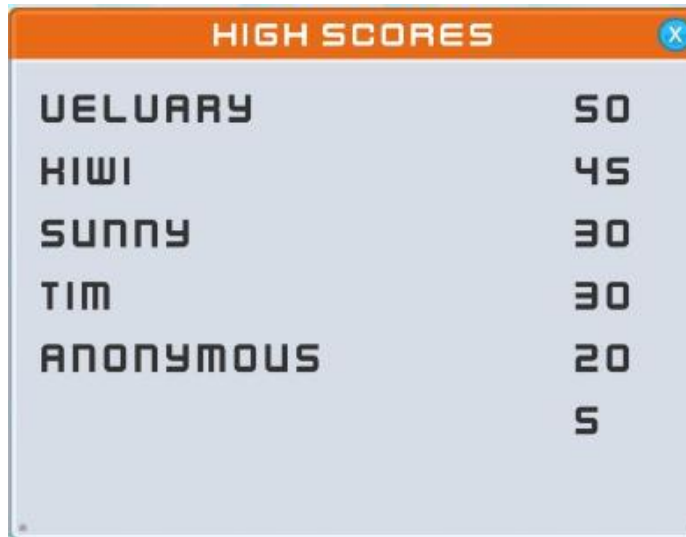


Hình 3.9. Vật phẩm bom




Hình 3.10. Mô tả đường bay đạn của súng ACR

- + **Health:**
  - Hiện thị lượng máu của nhân vật người chơi..
  - Thanh máu sẽ chuyển dần từ xanh lá khi đầy, dần chuyển sang vàng, cuối cùng chuyển sang đỏ khi máu ở ngưỡng nguy hiểm. .
  - Trạng thái khi bị nhận sát thương có hiệu ứng rung lắc nhẹ.
  - Khi nhân vật bị hết máu, thanh máu trở nên trống và trò chơi kết thúc.
- + **Thông số người chơi:**
  - Máu cơ bản: 100 máu.
  - Tốc độ di chuyển: 2.
  - Sát thương cơ bản: Tùy thuộc vào thông số sát thương của từng loại vũ khí.
  - Khả năng hồi sinh: Không.
  - Jumpforce (Lực nhảy): 150
- + **Scoring:**
  - Giá trị điểm số: Điểm cộng thêm sẽ gia tăng tùy thuộc vào số lượng quái vật người chơi giết liên tục trong 1 khoảng thời gian nhất định. Với mỗi quái vật khác nhau sẽ đem lại cho người chơi các giá trị điểm khác nhau.
    - Quái dơi: 100 điểm / 1 đơn vị bị tiêu diệt.
    - Quái nhím: 120 điểm / 1 đơn vị bị tiêu diệt.
    - Quái lính đánh thuê: 150 điểm / 1 đơn vị bị tiêu diệt.
  - Bảng thành tích tốt nhất: Hiện thị tên người chơi, thời gian vượt qua màn chơi nhanh nhất xếp theo thứ tự.






Hình 3.11. Tham khảo màn hình thành tích người chơi

- + Hệ thống cửa hàng: Người chơi có thể sử dụng tiền kiếm được để mua trang phục, các vật phẩm hỗ trợ như x2 vàng, tăng sát thương đạn...
  - Tăng số lượng máu khởi đầu: 400 vàng / 1 đơn vị máu, có giá trị vĩnh viễn. Kể từ sau lần đầu tiên, mỗi lần nâng cấp sau tăng thêm 40 vàng.
  - Tăng tốc độ di chuyển: 200 vàng / 1 tốc độ di chuyển, có giá trị vĩnh viễn. Kể từ sau lần đầu tiên, mỗi lần nâng cấp tăng thêm 100 vàng (Tối đa 10 tốc độ di chuyển).
  - x2 vàng: Giá bán 200 vàng, hiệu lực trong 3 ván thắng.
  - Tăng sát thương đạn - Tăng thêm 20% sát thương: Giá bán 300 vàng, hiệu lực trong 3 ván chơi.
  - Lá chắn: Giá bán 50 vàng, hiệu lực trong 1 màn chơi.
  - Trang phục: Giá bán trong khoảng 200 - 1000 vàng, có giá trị vĩnh viễn.
- + Major characters in story:

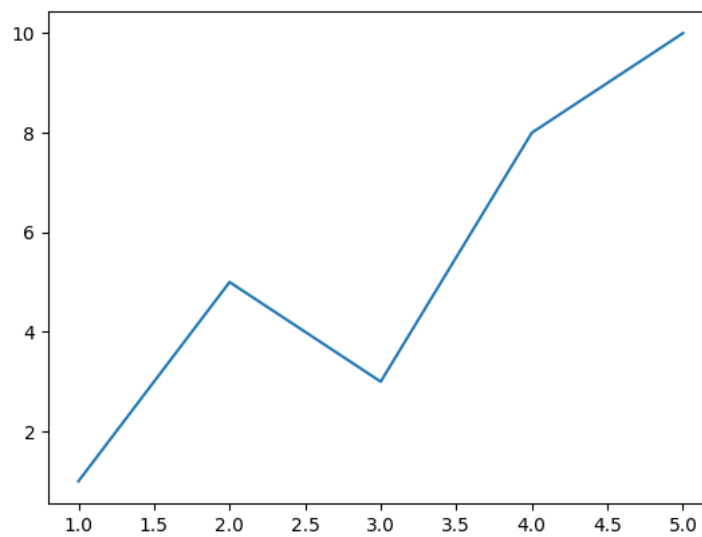
Nhân vật	Tên nhân vật	Mô tả
	Hogwart	Nhân vật đầu tiên và giúp đỡ xuyên suốt cho Michael trong trò chơi. Cung cấp thông tin cho Michael về cách chơi, các boss xuất hiện, quái vật, các map, chiến đấu.... Sử dụng các



		đòn đánh tay gậy sắt thương cho quái vật.
	Jasmine	Em gái của Hogwarts, chiến đấu cùng Michael và sử dụng cung tên làm vũ khí chính
	Timo	Sơn thần ở vùng núi, có 1 cuộc trò chuyện ngắn với Michael về vùng đất nơi đây
	Mario	Người may mắn thoát khỏi quái vật người phóng xạ, có cuộc trò chuyện ngắn với Michael

Bảng 3.2. Danh sách các nhân vật trong game

- + Game progression outline: Với mức khó cao nhất là đánh boss, những mức thấp hơn yêu cầu hoàn thành những nhiệm vụ với mức vừa và dễ.



Hình 3.12. Biểu đồ nhíp


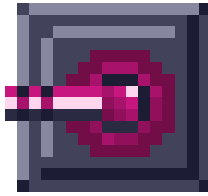

+ Game levels:

Level	Mô tả chung	Nhiệm vụ, phần thưởng	Kẻ thù chính	Độ khó
1	Hướng dẫn chiến đấu và qua màn.	Tiêu diệt 5 quái nhỏ + Thưởng: 50 vàng	Lính đánh thuê	1
2, 8, 15	- Tiêu diệt số lượng quái nhất định. - Giới hạn về vũ khí.	- Tiêu diệt 10/20/30 quái nhỏ - Đạt 1000/2000/5000 điểm + Thưởng: 100 vàng - Vũ khí có thể trang bị: 2	Lính đánh thuê, công vô định	1
3, 10, 16	- Tiêu diệt số lượng quái nhất định. - Giới hạn về thời gian.	-Tiêu diệt 10/15/20 quái nhím + Thưởng: 100 vàng - Thời gian: 2 phút	Lính đánh thuê	2

4, 9, 14, 17	<ul style="list-style-type: none"> <li>- Đạt mức điểm tối thiểu để vượt qua màn chơi.</li> <li>- Vượt qua màn chơi mà không bị mất máu</li> </ul>	<ul style="list-style-type: none"> <li>- Đạt 2000/4000/8000 điểm</li> <li>+Thưởng: 100 vàng</li> </ul>	Lính đánh thuê, ụ súng cố định, cổng vô định	2
5, 11, 18	<ul style="list-style-type: none"> <li>- Level 5: Đồng hành với Hogwart ở map khu rừng</li> <li>- Level 11: Đồng hành với Jasmine tại map sa mạc</li> <li>- Level 18: Đồng hành với Timo tại map núi lửa</li> </ul>	<ul style="list-style-type: none"> <li>- Tiêu diệt 30/40/50 quái</li> <li>+ Thưởng: 100 vàng.</li> </ul>	Lính đánh thuê, cổng vô định	3
6, 12, 19	<ul style="list-style-type: none"> <li>- Tiêu diệt số lượng quái nhất định.</li> <li>- Vượt qua màn chơi mà không ăn các vật phẩm hỗ trợ.</li> </ul>	<ul style="list-style-type: none"> <li>-Tiêu diệt 15/20/25 quái dơi.</li> <li>+ Thưởng: 100 vàng</li> </ul>	Lính đánh thuê, ụ súng cố định, cổng vô định	2
7, 13, 20	Các level đánh Boss tại các map (Khu rừng, núi lửa, đồi núi).	<ul style="list-style-type: none"> <li>- Tiêu diệt các boss</li> <li>+ Thưởng: 200/400/800 vàng và 10000/12000/15000 điểm</li> </ul>	Rồng ngàn tuổi, nhím chúa, người phóng xạ, lính đánh thuê, ụ súng cố định, cổng vô định	4

Bảng 3.3. Mô tả các level trong game

## + Quy tắc quái:

Quái	Tên quái	Mô tả
	Cổng vô định	Có khả năng sinh ra các quái lính đánh thuê tự động. Mỗi cánh cổng nếu tiêu diệt được sẽ nhận được 100 điểm  + Máu cơ bản: 15 đơn vị máu. + Thời gian sau mỗi lần sinh quái: 10 giây.
	Ụ súng cố định	Tự động bắn đạn từ xa khi phát hiện kẻ địch.. Nếu tiêu diệt được nó, bạn sẽ nhận được 120 điểm  + Sát thương cơ bản: 3. + Máu cơ bản: 8 đơn vị máu. + Thời gian sạc lại đạn: 1,5 giây.
	Lính đánh thuê	Tấn công từ xa. Nếu tiêu diệt được, bạn sẽ nhận được 150 điểm  + Sát thương cơ bản: 2. + Máu cơ bản: 12 đơn vị máu. + Thời gian sạc lại đạn: 2 giây.


Bảng 3.4. Bảng mô tả quy tắc quái trong game



## + Level-specific enemies:

- Level 7: Rồng ngàn tuổi - Một loài rồng đỏ khổng lồ với khả năng phun lửa rộng và tấn công kẻ địch bằng quyền trượng.

- Sát thương cơ bản: 3 - 6.
  - Máu cơ bản: 50 đơn vị máu.
  - Mỗi 3 đòn đánh thường sẽ có 1 lần phun lửa.
  - Giá trị sát thương: 5 đơn vị máu với đòn đánh thường và 12 đơn vị máu với đòn phun lửa.
  - Nếu tiêu diệt được nhận được 10000 điểm.
- Level 13: Nhím chúa - Loài nhím khổng lồ với khả năng tấn công bằng gai nhọn, có sức chống chịu cao. Gai nhọn có khả năng gây chảy máu nếu như người chơi trúng phải.
- Sát thương cơ bản: 4 - 8.
  - Máu cơ bản: 70 đơn vị máu.
  - Giá trị sát thương: 3 đơn vị máu với mỗi gai nhọn và 6 sát thương trong mỗi 4 giây nếu bị trúng độc.
  - Nếu tiêu diệt được nhím chúa sẽ nhận được 12000 điểm.
- Level 20: Người phóng xạ - Tên dị nhân với khả năng tấn công bằng súng laser liên tục sau một khoảng thời gian ngắn.
- Sát thương cơ bản: 5 - 7.
  - Máu cơ bản: 100 đơn vị máu.
  - Giá trị sát thương: 10 đơn vị máu với đòn đánh laser trúng phải.
  - Nếu tiêu diệt nhận được 15000 điểm.

+ Bosses:

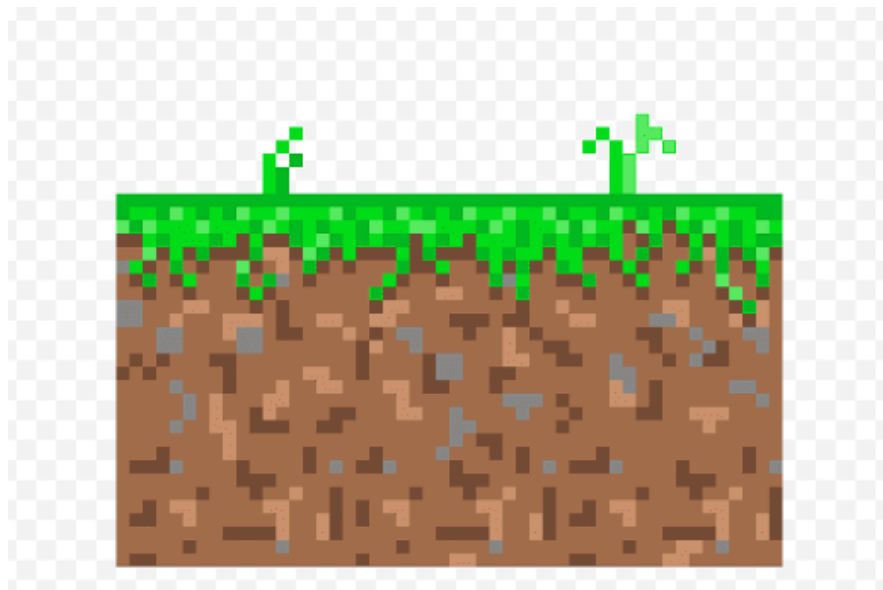
Boss	Tên boss	Mô tả
	Rồng ngàn tuổi	<p>- Rồng ngàn tuổi - Một loài rồng đỏ khổng lồ với khả năng phun lửa rộng và tấn công cận chiến kẻ địch bằng quyền trượng.</p> <p>- Mỗi 3 đòn cận chiến rồng sẽ vận sức trong 2s để phun ra lửa</p>

	<p>Nhím khổng lồ</p>	<ul style="list-style-type: none"> <li>- Nhím chúa - Loài nhím khổng lồ với khả năng tấn công bằng gai nhọn, có sức chống chịu cao.</li> <li>- Gai nhọn có khả năng gây chảy máu nếu như người chơi trúng phải</li> </ul>
	<p>Người phóng xạ</p>	<ul style="list-style-type: none"> <li>- Người phóng xạ - Tên dị nhân với khả năng tấn công bằng súng laser liên tục sau một khoảng thời gian ngắn.</li> <li>- Có tốc độ di chuyển nhanh, có khả năng triệu hồi các quái lính đánh thuê để hỗ trợ</li> </ul>

Bảng 3.5. Mô tả các boss trong game

+ NPC:

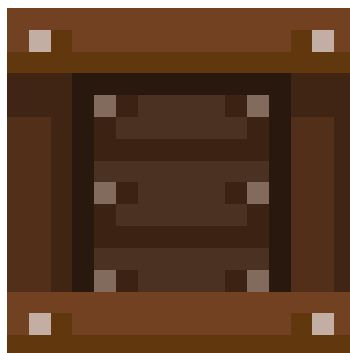
○ Môi trường:



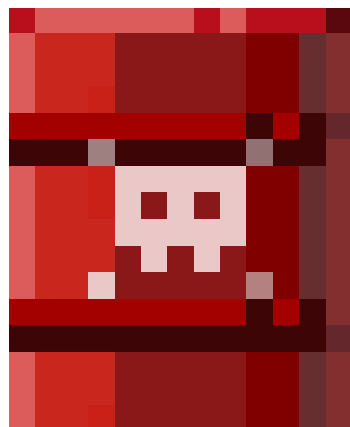
Hình 3.13. Địa hình trong game



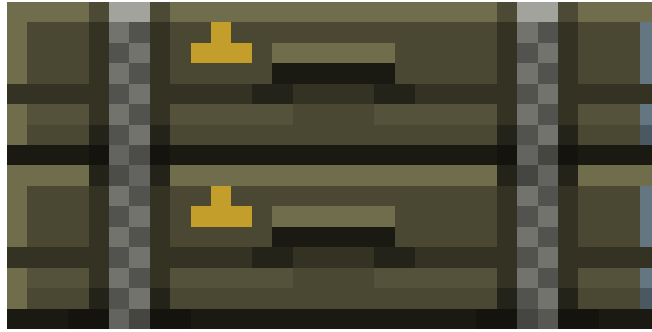
Hình 3.14. Môi trường trong game



Hình 3.15. Vật phẩm thùng gỗ nhỏ



Hình 3.16. Vật phẩm thùng thuốc nổ



Hình 3.17. Vật phẩm thùng gỗ lớn

- Vật phẩm hỗ trợ: Hồi máu, tăng tốc chạy, vàng.
- Nhân vật hỗ trợ, chiến đấu cùng Michael: Jasmine, Hogwart.
- Nhân vật phụ: Mario, Timo.
- + Cutscenes:
  - Cutscenes mở đầu khi vào game: Cảnh chiến đấu của Michael trong một ván đấu.
  - Cutscenes level 1: Michael gặp gỡ Hogwart lần đầu tiên. Hogwart giới thiệu với Michael về thế giới ảo mà Michael bị hút vào. Hogwart giải thích cho Michael về cơ chế chơi, hướng dẫn di chuyển và cách vượt qua màn chơi.
  - Cutscenes level 7: Michael trò chuyện cùng Hogwart. Hogwart thông báo cho Michael về thông tin boss rồng ngàn tuổi - một loài rồng với sát thương lớn và khả năng chống chịu cao tại vùng đất núi lửa, có thể bay được. Hogwart chúc Michael thành công vượt qua và mau chóng tiến tới vùng đồi núi.
  - Cutscenes level 13: Michael trò chuyện với Timo về 1 loài boss bò sát khổng lồ với đầy gai nhọn khổng lồ trên cơ thể. Sơn thần Timo nói về sự thống trị của loài nhím khổng lồ tại khu rừng và cảnh báo Michael về sự nguy hiểm của những chiếc gai nhọn.
  - Cutscenes level 20: Michael trò chuyện với Jasmine, Mario, Hogwart. Jasmine đã từng đối đầu với người phóng xạ trong quá khứ nhưng đã thất bại. Mario giải thích với Michael về khả năng của người phóng xạ và cách để đánh bại chúng. Hogwart thông báo đây là boss cuối cùng và nếu Michael đánh bại được nó, anh sẽ có cơ hội trở về thế giới thực.
  - Cutscenes kết thúc game: Michael tạm biệt Hogwart, Jasmine, Mario.
- + Music and SFX:



- Âm thanh nhạc nền khi chơi.
- Âm thanh khi vào game.
- Âm thanh khi vượt qua thử thách, thua cuộc.
- Âm thanh pause.
- Âm thanh khi ấn chơi lại.
- Âm thanh cảnh báo boss.
- Hiệu ứng tiếng súng, tiếng đạn.
- Âm thanh nhân vật khi bị gây sát thương.
- Âm thanh quái vật, tiếng đạn của quái vật ...
  - Lòng tiếng cho các nhân vật: Michael, Jasmine, Hogwart, Timo, Mario.
  - Âm thanh cho từng map cơ bản như núi lửa, rừng, sa mạc, ...

### **3.3. Technical Design Document**

#### **3.3.1. Danh sách các tính năng dựa trên GDD**

- Lựa chọn màn chơi: Người chơi có thể lựa chọn các level sau khi nhấn vào button start game.
- Điều khiển nhân vật chơi: Người chơi có thể điều khiển nhân vật bằng cách sử dụng các phím di chuyển W để nhảy lên, A để sang trái, D để sang phải, Space để bắn đạn.
- Thiết lập âm thanh: Người chơi có thể click vào button setting để điều chỉnh âm thanh và nhạc nền, cho phép họ bật/tắt theo nhu cầu.
- Xem highscore: Người chơi có thể click vào button highscore để xem số điểm cao nhất mà họ đã vượt qua qua các màn chơi.
- Thoát game: Người chơi có thể thực hiện đóng trò chơi khi họ không muốn chơi nữa.

#### **3.3.2. Các công cụ sử dụng**

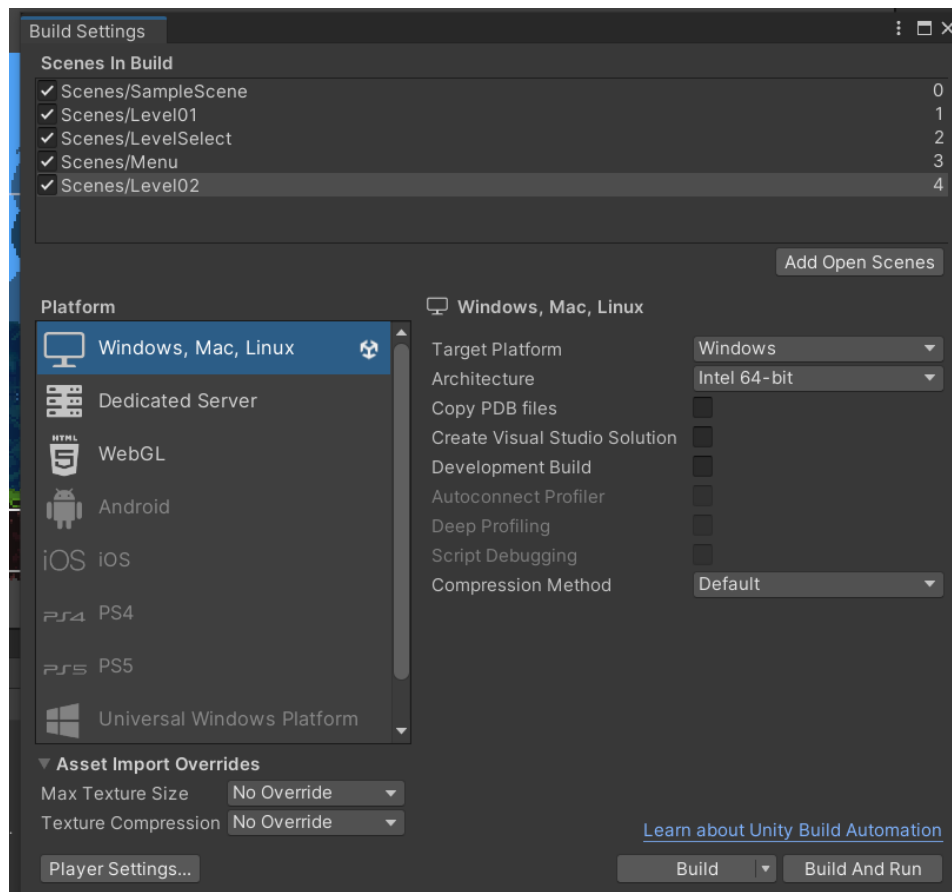
- Unity: Thiết kế đồ họa, xây dựng các scenes ...
- Visual Studio: Viết mã nguồn script trong game, mã logic trong Unity.
- Github: Tìm source code tham khảo, các thành viên đồng bộ hóa mã nguồn trên đó.
- Adobe Photoshop: Thiết kế logo game, các element trong trò chơi.

### 3.3.3. Bảng lịch trình

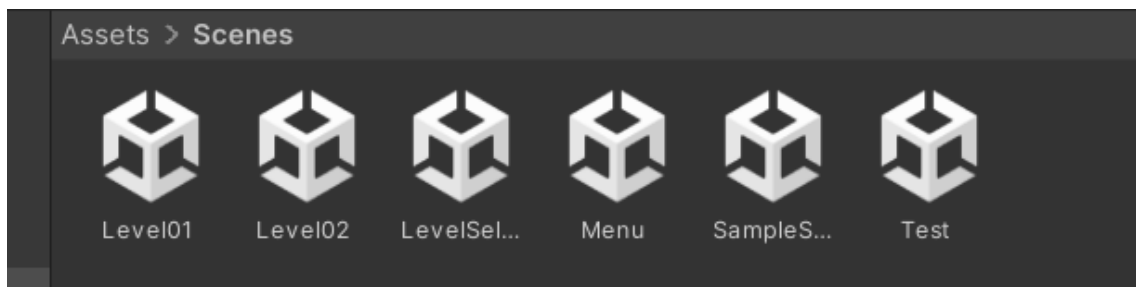
Các giai đoạn	Ngày bắt đầu	Ngày kết thúc	Nhiệm vụ	Người thực hiện	Mức độ hoàn thành
Tiền sản xuất	30/9/2023	2/10/2023	Tìm hiểu về công cụ	Quốc Đạt, Nguyễn Tuấn	100%
	2/10/2023	4/10/2023	Lựa chọn chủ đề	Quốc Đạt, Nguyễn Tuấn	100%
	4/10/2023	7/10/2023	Triển khai ý tưởng	Quốc Đạt, Nguyễn Tuấn	100%
	8/10/2023	8/10/2023	Họp nhóm phân chia công việc	Nguyễn Tuấn	100%
Sản xuất	9/10/2023	15/10/2023	Viết GDD	Nguyễn Tuấn	100%
	16/10/2023	20/10/2023	Thiết kế nhân vật Game	Hồng Nhung	100%
	21/10/2023	26/10/2023	Tham khảo texture, âm thanh	Quốc Đạt	100%
	27/10/2023	2/11/2023	Thiết kế giao diện game	Quốc Đạt	100%
	3/11/2023	20/11/2023	Thực hiện triển khai code màn hình bắt đầu, màn hình chọn level, setting ...	Quốc Đạt, Nguyễn Tuấn	100%
	21/11/2023	30/11/2023	Thực hiện triển khai code các level khi chơi	Quốc Đạt, Nguyễn Tuấn	100%
	1/12/2023	6/12/2023	Kiểm thử sản phẩm và sửa đổi nếu phát hiện lỗi	Quốc Đạt, Nguyễn Tuấn	100%
Hậu sản xuất	7/12/2023	10/12/2023	Phát hành bản dùng thử	Hồng Nhung, Nguyễn Tuấn	100%
	10/12/2023	15/12/2023	Họp tổng kết đánh giá hiệu quả	Hồng Nhung	100%

Hình 3.18. Bảng lịch trình triển khai kế hoạch phát triển game

### 3.3.4. Quản lý scene



Hình 3.19. Build Settings



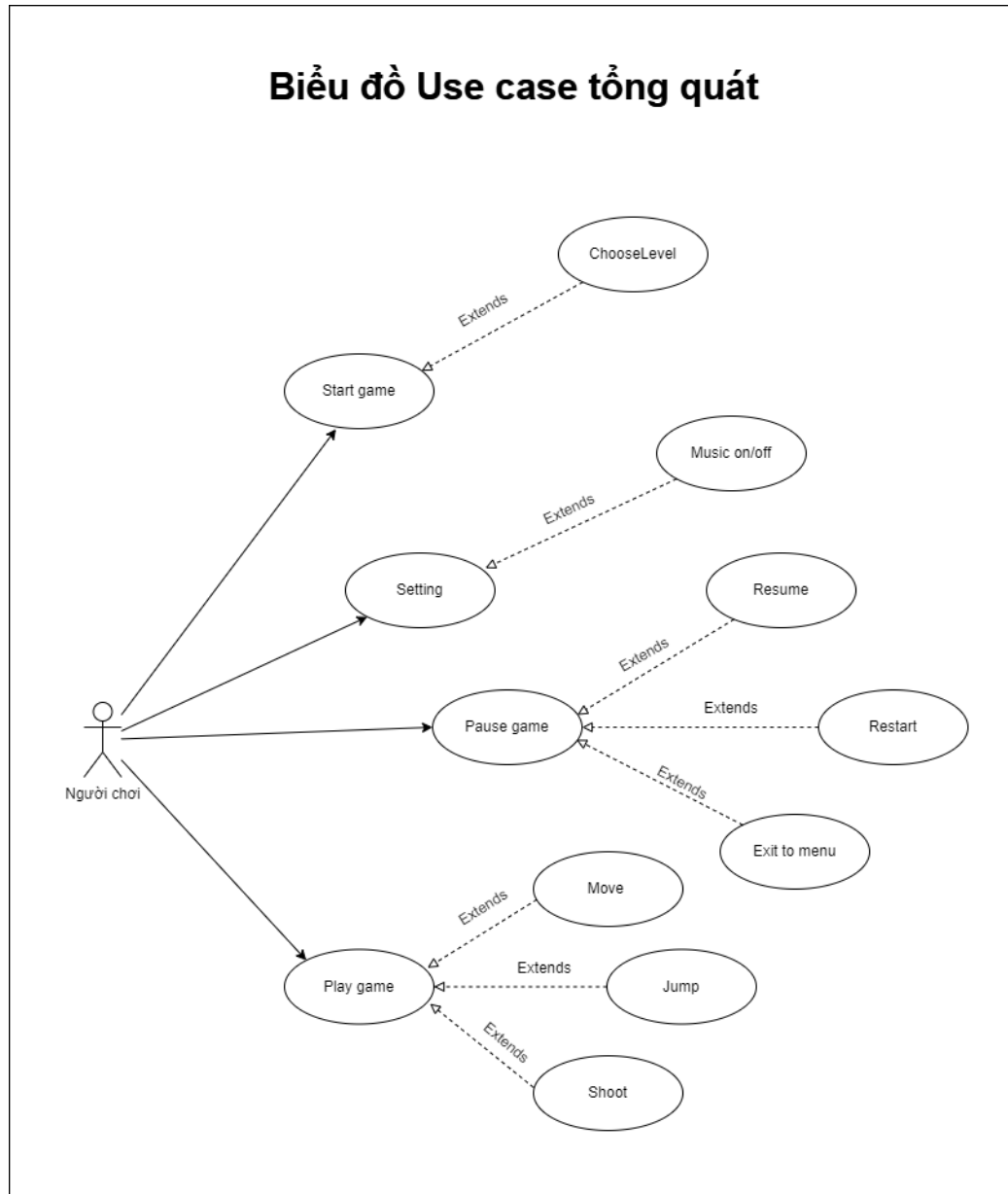
Hình 3.20. Quản lý các scene

### 3.3.5. Nền tảng, yêu cầu phần cứng và phần mềm

- Phần cứng:
  - + Bộ xử lý: Dual-core hoặc tương đương.
  - + Bộ nhớ RAM tối thiểu: 4GB RAM trở lên.
  - + Card đồ họa (GPU): Có hỗ trợ DirectX10.
  - + Ổ lưu trữ: Tối thiểu 5GB không gian trống
- Phần mềm:
  - + Hệ điều hành: Windows 7 trở lên (đề xuất Windows 10) hoặc macOS.

### 3.4. Phân tích và thiết kế hệ thống

#### 3.4.1. Biểu đồ Usecase



Hình 3.21. Biểu đồ Usecase tổng quát

#### 3.4.2. Kịch bản (Scenario)

Usecase	Start Game
Tác nhân	Người chơi

Điều kiện trước	Người chơi ở màn hình Home
Điều kiện đảm bảo thành công	Màn hình hiển thị màn hình chọn màn chơi
Chuỗi sự kiện chính: <ol style="list-style-type: none"> <li>1. Người chơi click vào button Play.</li> <li>2. Màn hình “Home” ẩn đi, chuyển sang màn hình chọn màn chơi.</li> </ol>	
Ngoại lệ	Không

Bảng 3.6. Đặc tả use case start game

Usecase	Choose Level
Tác nhân	Người chơi
Điều kiện trước	Người chơi đã click vào button Play
Điều kiện đảm bảo thành công	Màn hình chọn các màn chơi chuyển sang màn hình Gameplay và hiển thị level đã chọn
Chuỗi sự kiện chính: <ol style="list-style-type: none"> <li>1. Người chơi click vào 1 level được hiển thị trên màn hình chọn màn chơi sau khi đã click button play ở màn home.</li> <li>2. Màn hình chuyển qua màn hình gameplay và hiển thị level đã chọn.</li> </ol>	
Ngoại lệ	Không

Bảng 3.7. Đặc tả use case choose level

Usecase	Setting
Tác nhân	Người chơi

Điều kiện trước	Người chơi đang ở màn hình Home
Điều kiện đảm bảo thành công	Màn hình hiển thị giao diện setting
Chuỗi sự kiện chính: <ol style="list-style-type: none"> <li>1. Người chơi click vào button Setting ở màn hình home</li> <li>2. Màn hình home hiển thị giao diện setting</li> </ol>	
Ngoại lệ	Không

Bảng 3.8. Đặc tả use case setting

Usecase	Bật tắt sound và music
Tác nhân	Người chơi
Điều kiện trước	Người chơi click vào button setting ở màn hình home và hiển thị giao diện setting
Điều kiện đảm bảo thành công	Game bật hoặc tắt nhạc và âm thanh theo nhu cầu của người chơi
Chuỗi sự kiện chính: <ol style="list-style-type: none"> <li>1. Người chơi click vào button loa hoặc hình nốt nhạc trên giao diện setting</li> <li>2. Hệ thống sẽ thực hiện dựa vào button mà người chơi vừa ấn để bật nhạc hoặc tắt âm thanh, nhạc của trò chơi</li> </ol>	
Ngoại lệ	Không

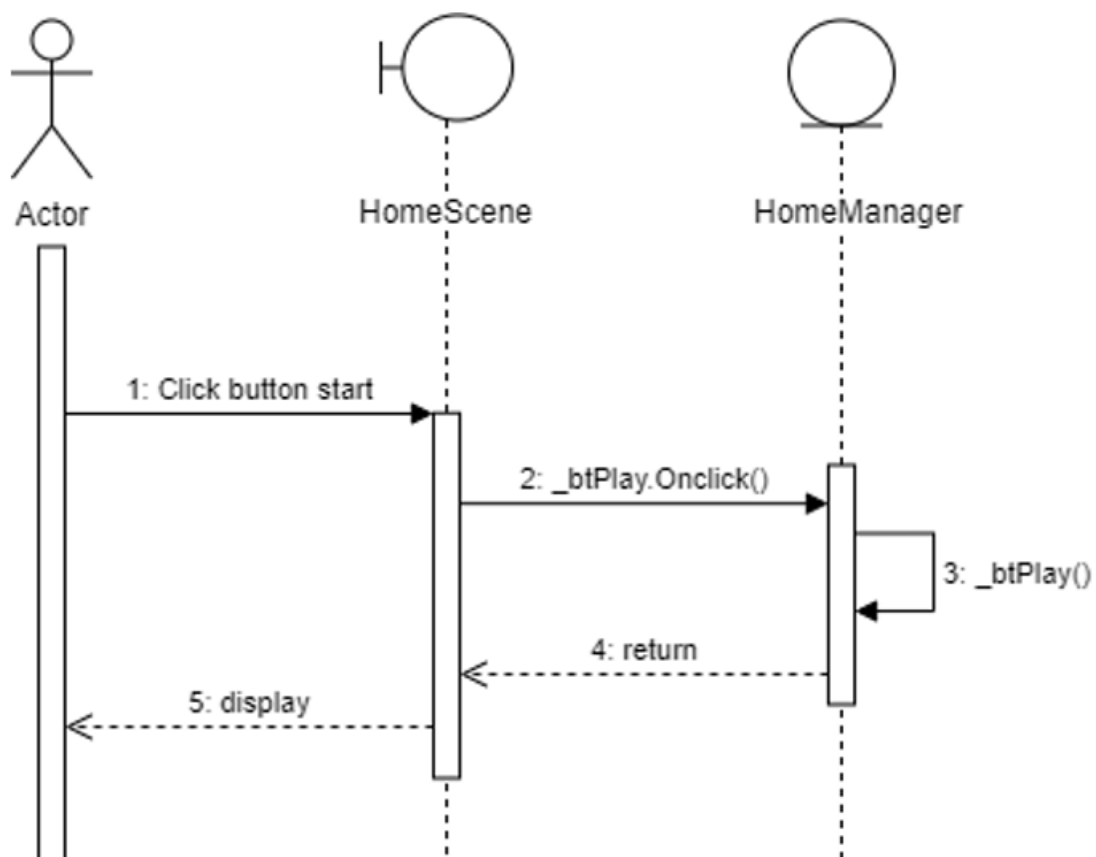
Bảng 3.9. Đặc tả use case bật tắt sound và music

Usecase	CharacterController
Tác nhân	Người chơi

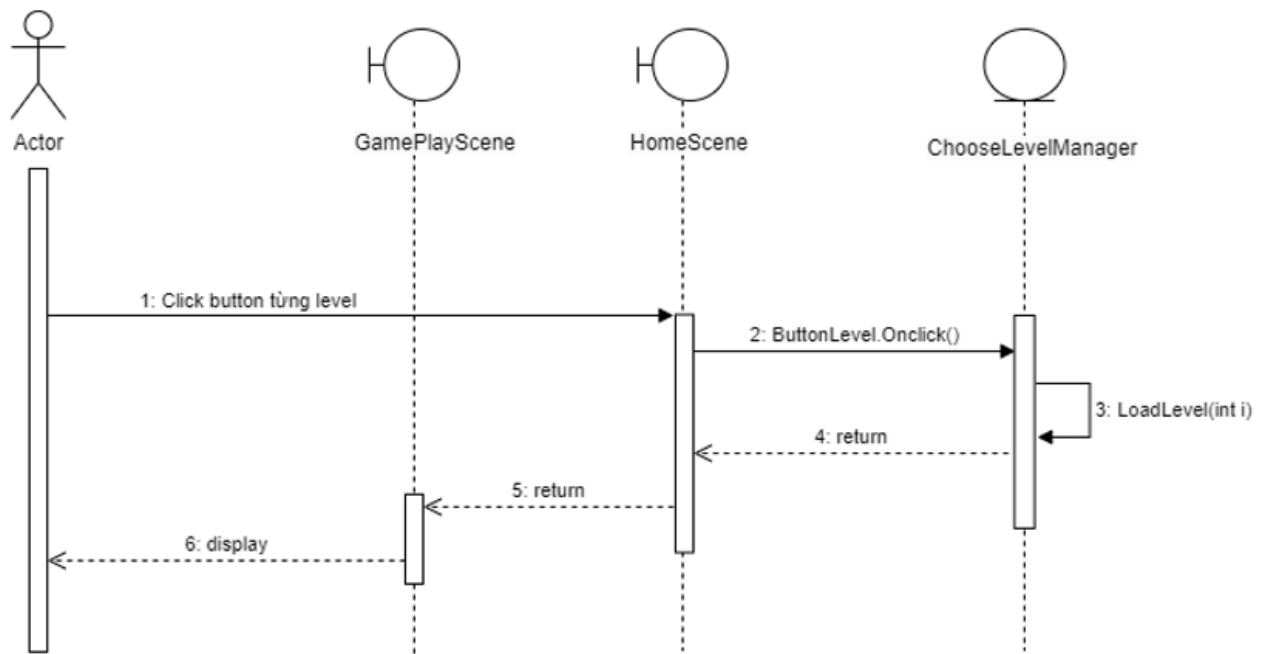
Điều kiện trước	Người chơi vào được màn hình Gameplay
Điều kiện đảm bảo thành công	Nhân vật di chuyển, nói cách khác thực hiện theo điều khiển của người chơi
Chuỗi sự kiện chính: <ol style="list-style-type: none"> <li>1. Người chơi sử dụng các phím W, A, D, Space hoặc B trên bàn phím</li> <li>2. Nhân vật sẽ di chuyển theo điều khiển của người chơi và hiển thị chuyển động đó trên màn hình chơi.</li> </ol>	
Ngoại lệ	Không

Bảng 3.10. Đặc tả use case character controller

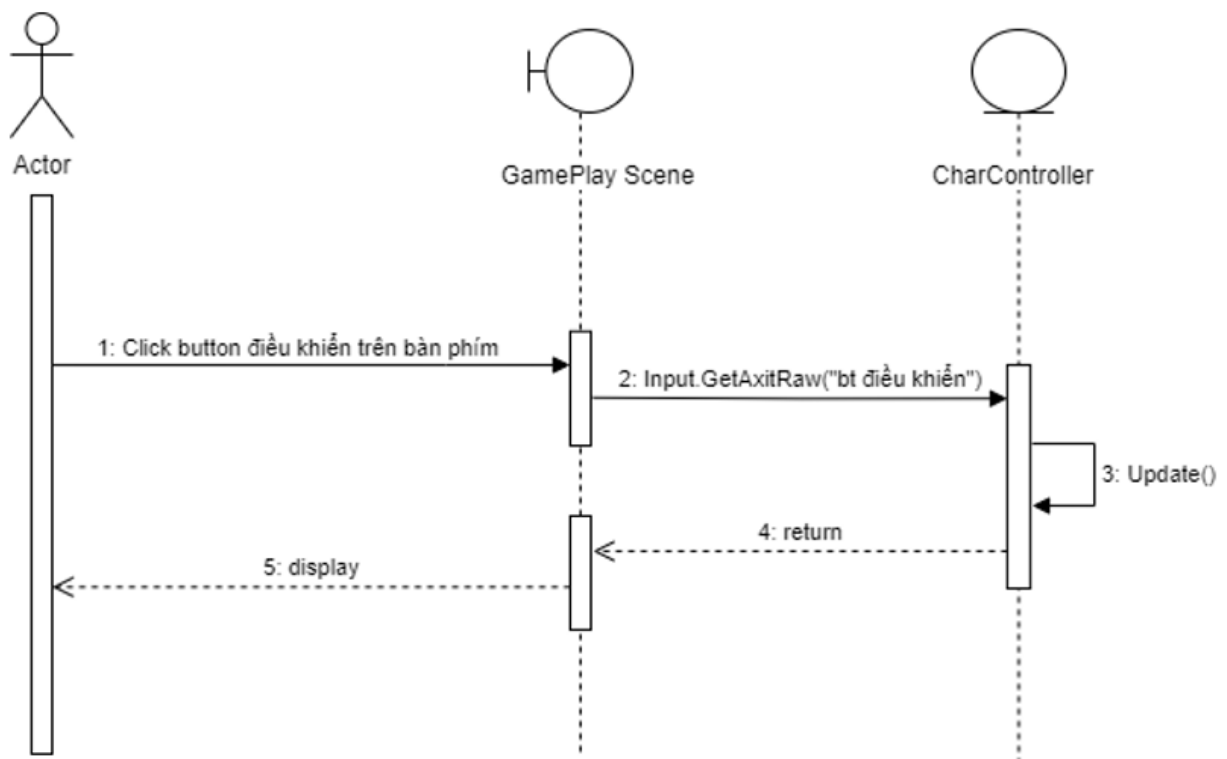
### 3.4.3. Biểu đồ tuần tự (Sequence)



Hình 3.22. Biểu đồ tuần tự chức năng start game

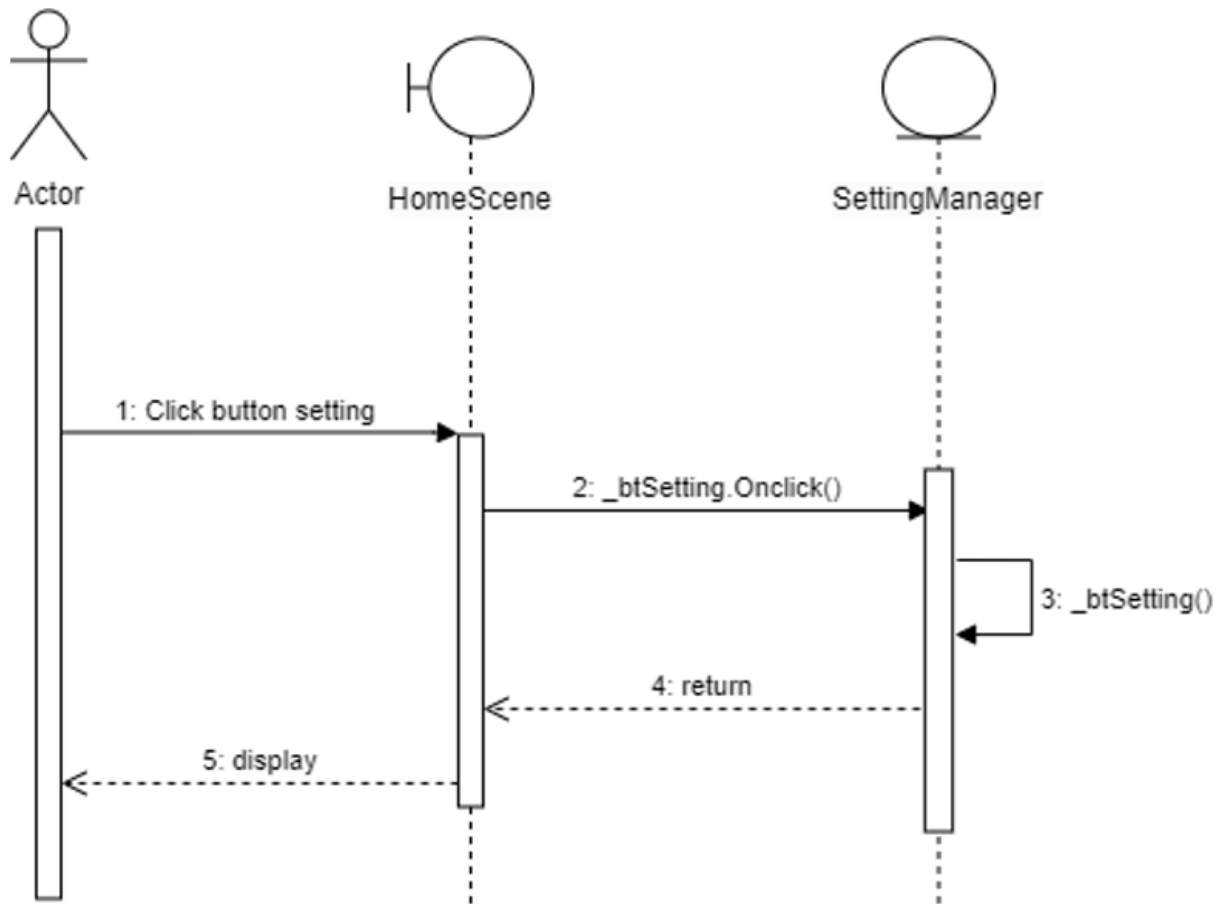


Hình 3.23. Biểu đồ tuần tự chức năng choose level

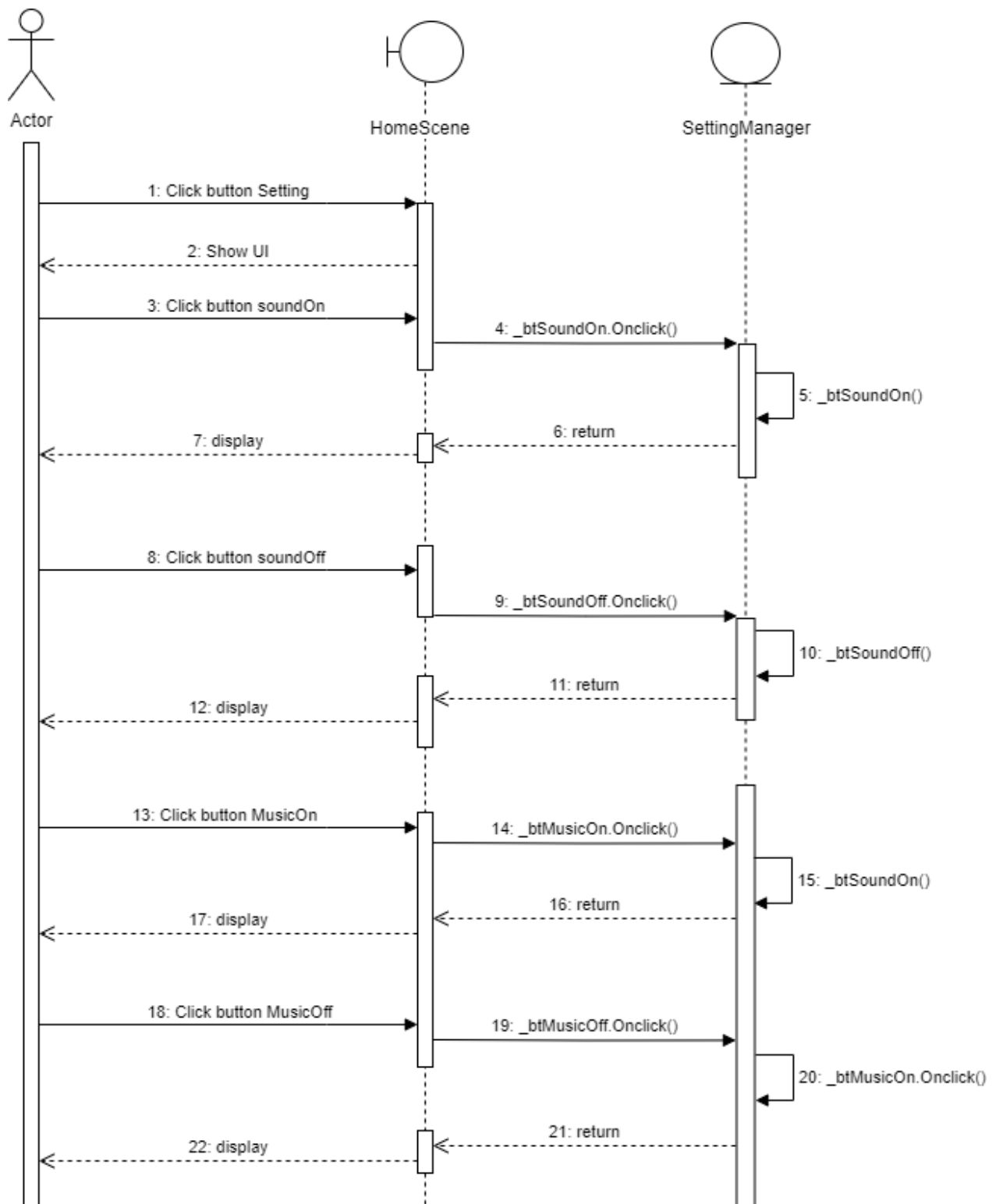


Hình 3.24. Biểu đồ tuần tự chức năng di chuyển của nhân vật



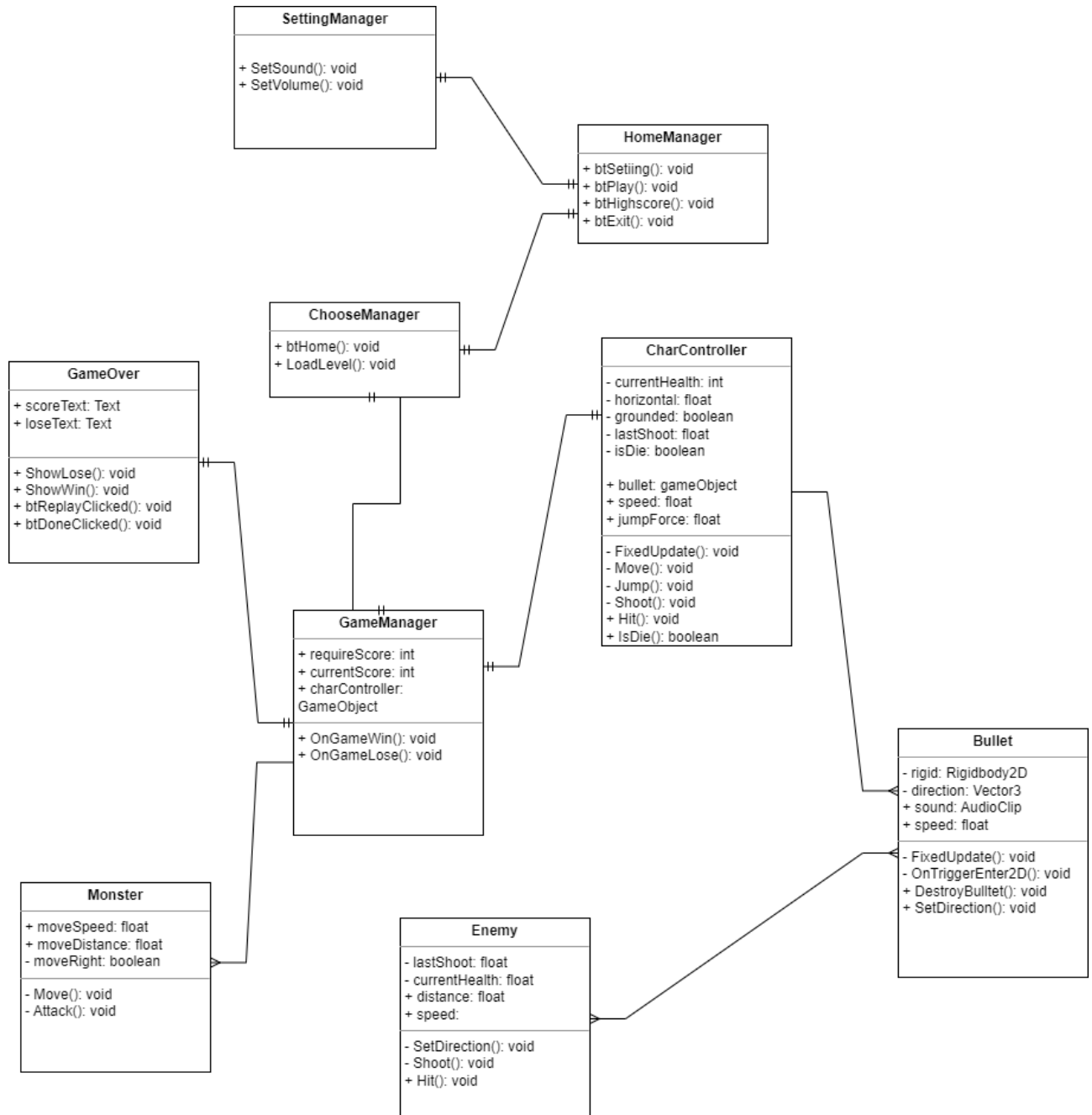


Hình 3.25. Biểu đồ tuần tự chức năng hiển thị giao diện setting



Hình 3.26. Biểu đồ tuần tự chức năng setting

#### 3.4.4. Biểu đồ lớp thiết kế



Hình 3.27. Biểu đồ lớp thiết kế

### 3.5. Tiểu kết chương 3

Chương 3 đã giúp định rõ mục tiêu và ý nghĩa, cũng như sự cấp thiết của việc phát triển trò chơi "Run or Die". Nhóm đã không chỉ thực hiện một cái nhìn tổng quan về trò chơi mà còn giới thiệu chi tiết về tài liệu thiết kế game (GDD) mà nhóm đã công phu phát triển. Đây là một bước quan trọng để chúng em duy trì sự nhất quán và tiến triển theo đúng

kế hoạch đã đề ra. Việc này không chỉ giúp đảm bảo rằng dự án sẽ tuân thủ theo các mô tả và chi tiết trong phân tích và thiết kế hệ thống, mà còn giúp duy trì sự đồng thuận trong nhóm làm việc.

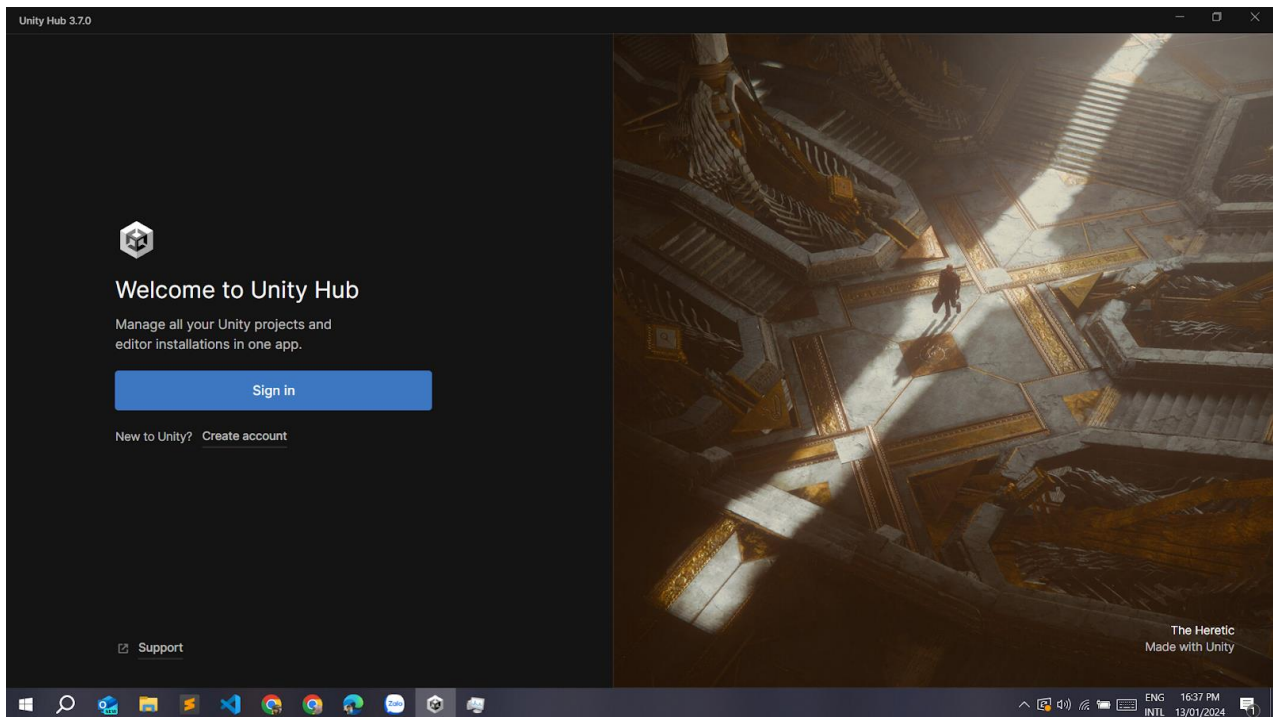
Trong chương tiếp theo, chúng em sẽ giới thiệu về quá trình cài đặt và triển khai hệ thống của trò chơi. Chúng em cũng sẽ đề cập đến việc thiết kế các màn chơi, nơi mà người chơi sẽ trải nghiệm những thử thách và hấp dẫn đặc sắc của "Run or Die". Bằng cách này, chúng em hy vọng sẽ chia sẻ với độc giả một cái nhìn chi tiết và rõ ràng nhất về quá trình phát triển và hoàn thiện trò chơi của nhóm. Hãy cùng nhau đồng hành và khám phá hành trình tạo ra "Run or Die"!

## Chương 4. CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG

### 4.1. Cài đặt, triển khai hệ thống

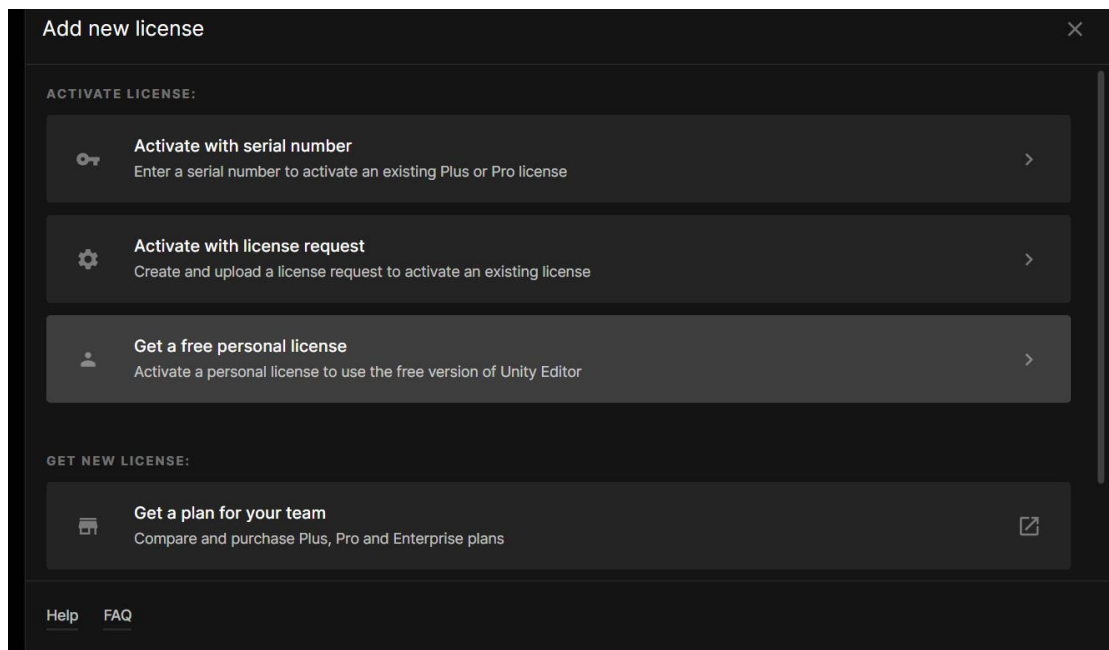
#### 4.1.1. Cài đặt công cụ Unity

- B1: Truy cập trang web <https://unity.com/download> để tải file cài đặt về máy.
- B2: Thực hiện cài đặt Unity về máy, sau đó ta thực hiện đăng nhập/tạo tài khoản nếu chưa có. Ta có thể tạo bằng Facebook/Google/Apple Account hoặc số điện thoại.



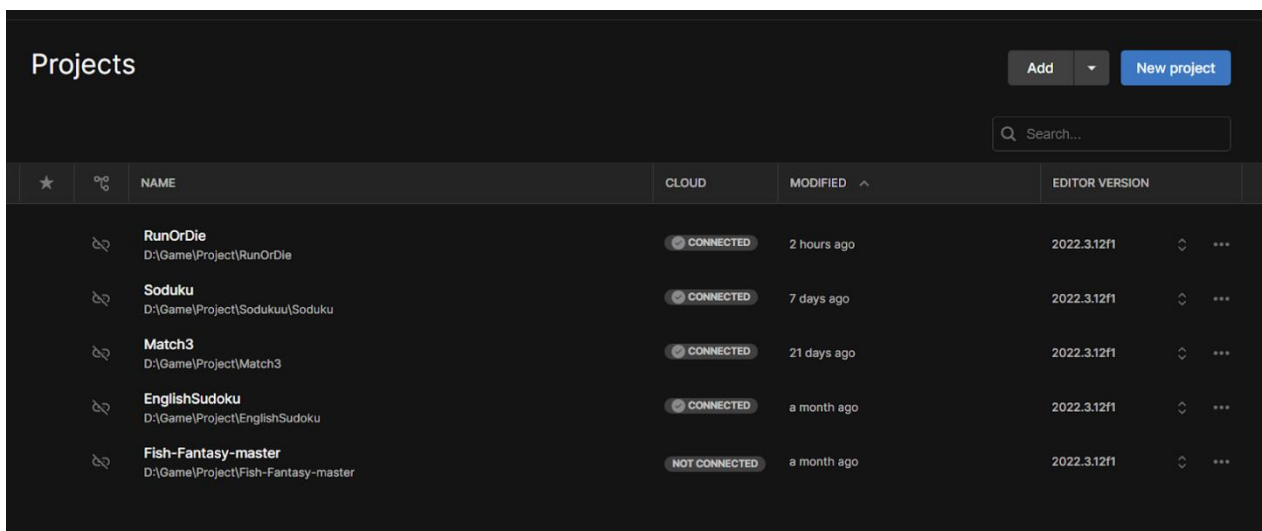
Hình 4.1. Màn hình download Unity

- B3: Kích hoạt giấy phép sử dụng: Ta lựa chọn loại dành cho cá nhân, có thời hạn trong vòng 1 năm.



Hình 4.2. Lựa chọn giấy phép sử dụng

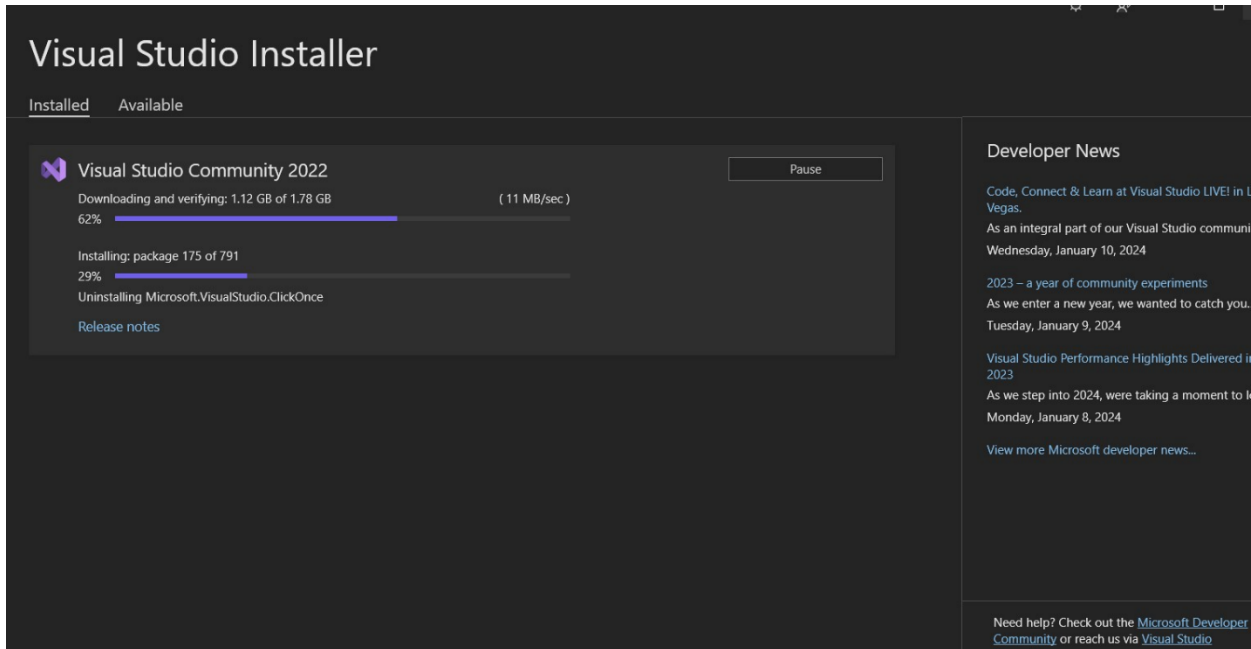
- B4: Cài đặt Unity Editor Phiên bản lựa chọn là bản 2022.3.12f1. Sau khi cài đặt thành công, các project sẽ được hiển thị ở đây. Ta có thể tạo mới hoặc import những project đã có sẵn trên máy.



Hình 4.3. Giao diện Unity sau khi cài đặt thành công

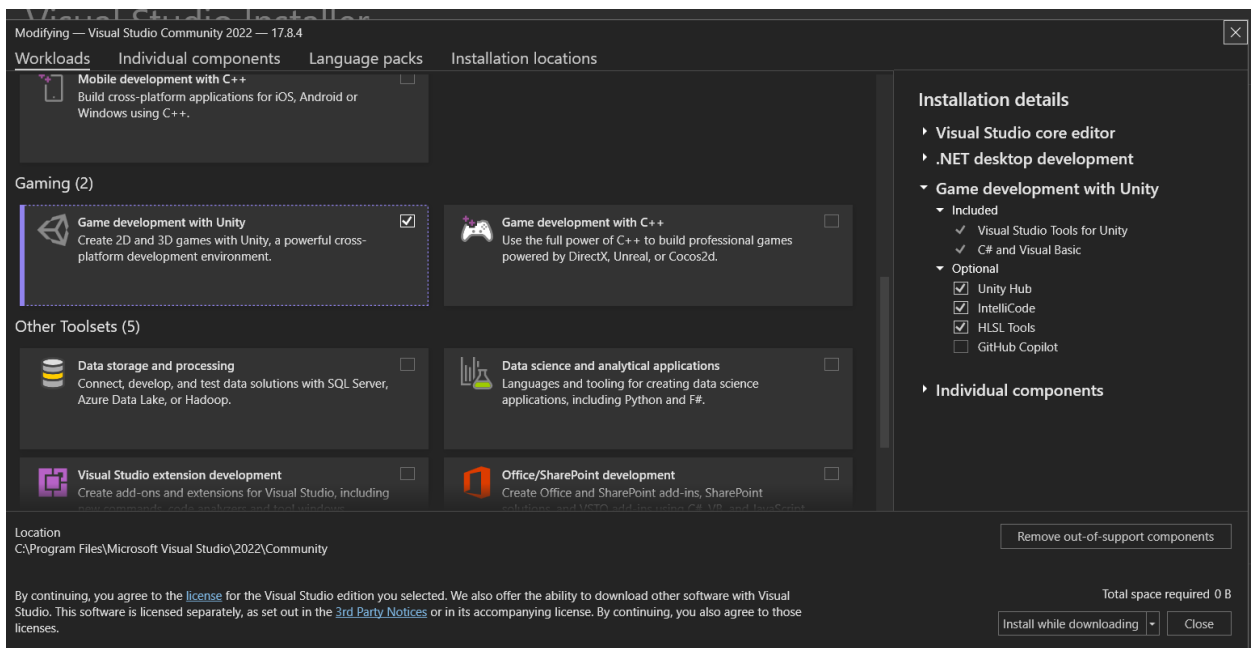
#### 4.1.2. Cài đặt IDE Visual Studio 2022

- B1: Truy cập trang web: <https://visualstudio.microsoft.com/downloads/> để tải file cài đặt về máy.
- B2: Thực hiện cài đặt Visual Studio Installer.



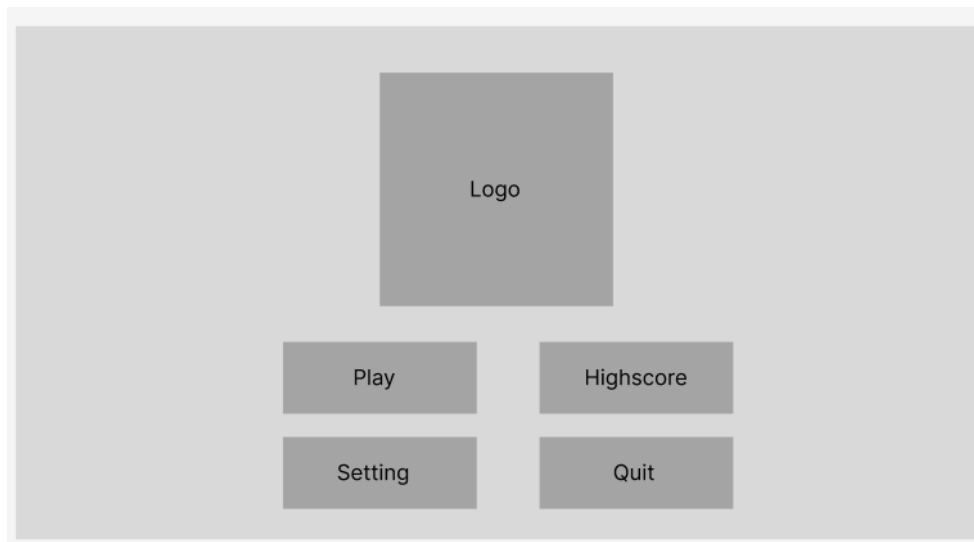
Hình 4.4. Giao diện cài đặt Visual Studio

- B3: Modifying lựa chọn Game Development with Unity.

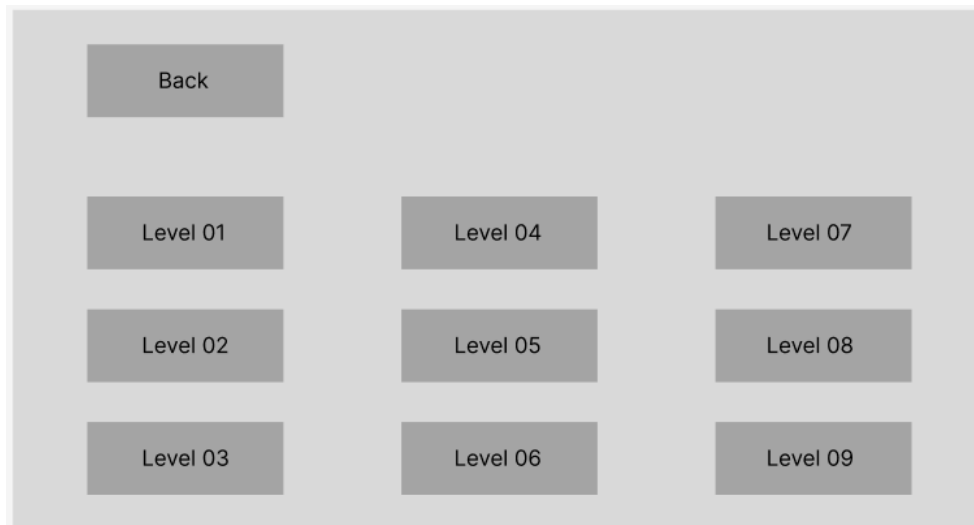


Hình 4.5. Lựa chọn game development với unity

#### 4.1.3. Thiết kế wireframe



Hình 4.6. Thiết kế wireframe giao diện homescreen



Hình 4.7. Thiết kế wireframe giao diện levelselect





Hình 4.8. Thiết kế wireframe giao diện gameplay



Hình 4.9. Thiết kế wireframe giao diện game over

#### 4.1.4. Một số màn hình chính trong game

- HomeScreen: Màn hình gồm 4 button Play, Highscore, Setting, Quit.
  - + Play: Cho người dùng vào giao diện LevelSelect để chọn màn chơi
  - + HighScore: Hiện thị bảng điểm cao nhất người chơi đạt được trong các màn chơi.
  - + Setting: Cho phép người chơi cài đặt bật tắt âm thanh hoặc nhạc nền của trò chơi.
  - + Quit: Cho phép người chơi thoát khỏi trò chơi.



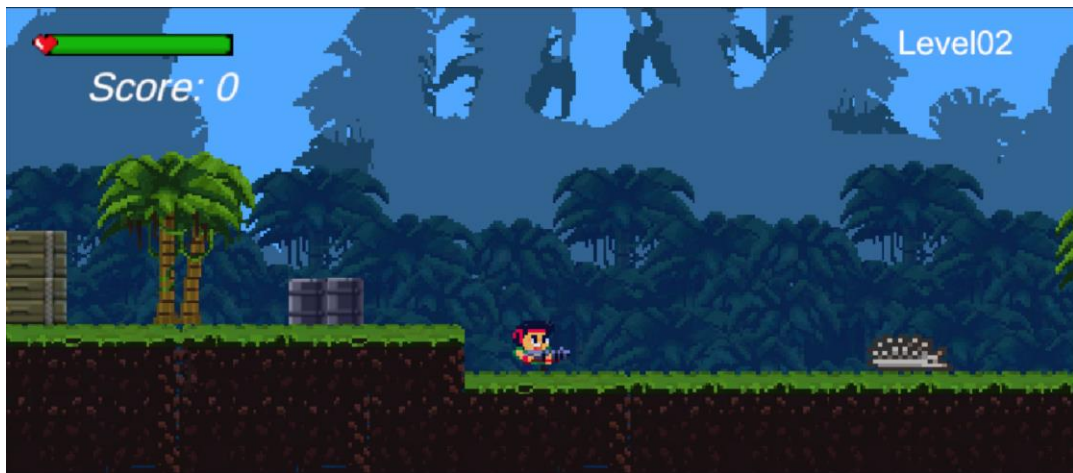
Hình 4.10. Giao diện home screen

- Level Select:
  - + Gồm 1 danh sách các level.
  - + Level chỉ hiển thị khi người chơi đã vượt qua màn chơi trước đó.



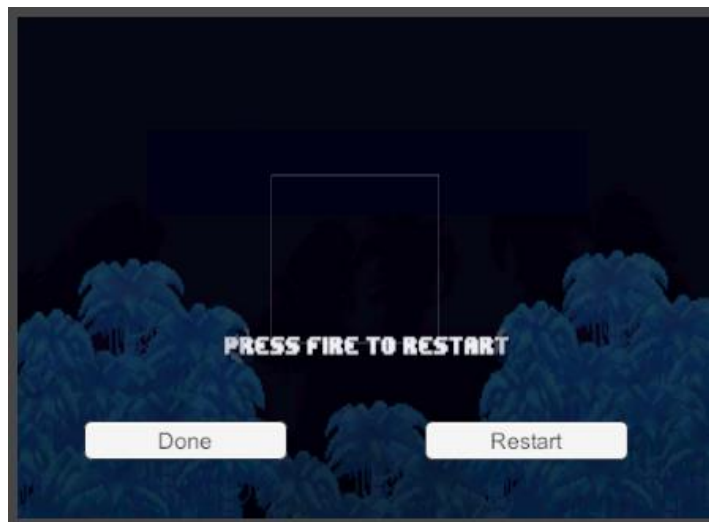
Hình 4.11. Giao diện level select

- GamePlay:
  - + Giao diện màn chơi gồm thanh máu của nhân vật ở góc trái trên màn hình.
  - + Thanh máu của từng kẻ thù hiển thị trên đầu của chúng.



Hình 4.12. Giao diện game play

- GameOver(Game Win/Lose):
  - + Khi người chơi vượt qua màn chơi nhờ đạt số điểm yêu cầu hoặc thua do bị hết máu, 1 giao diện chạy xuống hiển thị “You Win” hoặc “You Lose” thông báo cho người chơi.
  - + Button Restart: Cho phép người chơi chơi lại màn chơi đó.
  - + Button Done: Chuyển người dùng về giao diện LevelSelect để tiếp tục lựa chọn màn chơi.



Hình 4.13. Giao diện game over

#### 4.1.5. Thiết kế màn chơi trong game

- Mục tiêu: Đặt người chơi vào nhân vật Michael sinh tồn trong một không gian với kẻ địch săn lùng. Người chơi phải giết được số lượng quái theo yêu cầu, đạt được số điểm tối thiểu để có thể vượt qua được màn chơi đó.

- Cấu trúc bản đồ: Người chơi bắt đầu ở điểm đầu khu rừng, bên trong khu rừng đầy rẫy những kẻ thù, quái rừng, và những thứ nguy hiểm khác.
- Môi trường: Khu rừng toàn cây cối, chướng ngại vật, nhân vật chính, quái lính đánh thuê, quái nhím.
  - + Thời gian: Ban ngày
  - + Các yếu tố khác: Cây cỏ, thùng rỗng ...
- Mô tả logic trong trò chơi:
  - + Nhiệm vụ: Người chơi phải tiêu diệt số lượng kẻ thù nhất định để có thể đạt được số điểm yêu cầu để vượt qua màn chơi.
  - + Người chơi điều khiển nhân vật Michael di chuyển từ đầu đến cuối map, sử dụng súng để tiêu diệt hết kẻ thù.

#### 4.2. Đánh giá kết quả

- Đánh giá kết quả:

STT	Tiêu chí	Ưu điểm	Nhược điểm
1	Đồ họa	- Đồ họa sắc nét, tạo cảm giác sống động, chân thực. Giao diện người dùng thân thiện, dễ sử dụng	- Một số thiết bị có cấu hình thấp có thể xảy ra tình trạng giật lag.
2	Gameplay	- Đáp ứng được nhu cầu chơi game cơ bản. - Hướng dẫn chơi game rõ ràng giúp người mới không mất quá nhiều thời gian để làm quen với trò chơi.	- Hạn chế về màn chơi do khả năng phát triển của các thành viên trong nhóm phát triển. Màn chơi chưa đủ độ khó, có thể gây mất hứng thú với người chơi. - Chưa có nhiều vũ khí cho người chơi lựa chọn, chưa có cửa hàng mua đồ.
3	Trải nghiệm người dùng	- Người chơi tương tác mượt mà, không gặp lỗi quá nghiêm trọng trong quá trình trải nghiệm trò chơi.	- Người chơi có thể gặp khó khăn khi lần đầu trải nghiệm người chơi do chưa quen các phím điều khiển.

			<ul style="list-style-type: none"> <li>- Một số lỗi và bug có thể ảnh hưởng đến trải nghiệm chơi.</li> <li>- Thời gian chờ lâu khi load game có thể giảm đi sự hứng thú của người chơi.</li> </ul>
4	Âm thanh	<ul style="list-style-type: none"> <li>- Phù hợp với nội dung, tăng cường trải nghiệm cho người chơi.</li> </ul>	<ul style="list-style-type: none"> <li>- Âm thanh lặp lại có thể khiến người chơi thấy khó chịu, làm giảm chất lượng trải nghiệm người chơi.</li> <li>- Âm thanh không đồng bộ với hình ảnh hoặc gameplay có thể gây ra sự không nhất quán</li> </ul>
5	Các thông số thiết lập trong game	<ul style="list-style-type: none"> <li>- Game chạy với FPS ổn định (30FPS), tạo ra trải nghiệm người chơi mượt mà và không có hiện tượng giật lag</li> <li>- Người chơi dễ dàng điều khiển và tham gia vào các hoạt động trong game</li> </ul>	<ul style="list-style-type: none"> <li>- Trò chơi yêu cầu một phần cứng trung bình đến cao để đạt được FPS ổn định, điều này có thể là một thách thức đối với các thiết bị có cấu hình thấp.</li> </ul>

Bảng 4.1. Đánh giá kết quả

- Hướng phát triển tiếp theo:
  - + Phát triển đa dạng hơn các màn chơi: Nhóm sẽ tập trung vào việc phát triển một loạt các màn chơi mang tính đa dạng cao, không chỉ về số lượng mà còn về chất lượng. Chúng em sẽ xây dựng các màn chơi với nhiệm vụ và mức độ thách thức cao hơn, nhằm tránh tình trạng nhàm chán cho người chơi và tạo động lực cho họ khám phá những điều mới mẻ trong trò chơi.
  - + Phát triển thêm các yếu tố phụ trong game: Nhóm sẽ mở rộng thế giới game bằng cách thêm vào nhiều yếu tố phụ hấp dẫn. Điều này có thể bao gồm sự xuất hiện

của quái vật đa dạng, những vũ khí mới, các chương ngại vật đặc biệt, cũng như các nhiệm vụ phụ để thách thức người chơi và tạo ra trải nghiệm chơi game đa dạng.

- + Hoàn thiện các chức năng còn thiếu: Nhóm sẽ tập trung vào việc hoàn thiện các chức năng chưa đầy đủ trong trò chơi. Điều này có thể bao gồm triển khai chức năng highscore, xây dựng hệ thống setting linh hoạt, khả năng restart dễ dàng và phát triển hệ thống cửa hàng để người chơi có thể mua sắm và nâng cấp vũ khí và trang phục của mình.

#### **4.3. Tiểu kết chương 4**

Như vậy qua chương 4, chúng em đã trình bày cơ bản hoàn thiện các giao diện màn hình chính trong trò chơi cũng như thiết kế, trực quan hóa các giao diện của ứng dụng. Việc thiết kế các màn chơi và các màn hình chính trong trò chơi đóng vai trò quan trọng trong việc tạo ra trải nghiệm người chơi độc đáo và cuốn hút. Các kết quả được đánh giá đảm bảo đáp ứng được mong đợi của người chơi cũng như nhà phát triển.

## KẾT LUẬN

Như vậy, trên đây là bài báo cáo cuối kỳ của nhóm 6 chúng em về bài tập lớn của môn học. Trò chơi “Run or Die” đã đạt được mức độ phát triển khá ấn tượng, đạt tới 80% (4/5 chức năng được hoàn thiện) so với mô tả trong tài liệu thiết kế game (GDD). Trong quá trình phát triển, nhiều chức năng quan trọng đã được triển khai và hoạt động một cách ổn định, mang lại cho người chơi trải nghiệm tích cực và hấp dẫn. Sự hỗ trợ của Unity Engine đã giúp chúng em tối ưu hóa quá trình phát triển, đồng thời làm nổi bật những ý tưởng sáng tạo của nhóm. Tuy nhiên, chúng em cũng thấu hiểu rằng vẫn còn một số chức năng chưa hoàn thiện hoặc gặp khó khăn do hạn chế về kỹ năng và thời gian của các thành viên trong nhóm. Điều này làm cho chúng em nhận thức được sự quan trọng của việc liên tục cải tiến và phát triển dự án của mình. Bài báo cáo này, mặc dù đã cố gắng hoàn thiện nhất có thể, vẫn tồn tại một số khuyết điểm. Chúng em mong nhận được sự thông cảm và lời góp ý chân thành của thầy để có thể nắm bắt được những điểm cần cải thiện và phát triển trong tương lai. Nhóm rất mong được nhận những ý kiến đóng góp xây dựng từ phía độc giả, điều này sẽ giúp nâng cao chất lượng của trò chơi "Run or Die" và đồng thời phát triển kỹ năng cá nhân của từng thành viên trong nhóm.

## PHỤ LỤC

### File code JohnMovement.cs:

*Hàm Update()*: Cập nhật trạng thái và hành động của nhân vật.

- Trong hàm có gọi đến phương thức Die() - nhằm xác định xem nhân vật đã chết hay chưa.
- Horizontal - giá trị đầu vào từ trục ngang (là phím mũi tên sang trái/phải hoặc chữ A/D), giá trị này sẽ được sử dụng để di chuyển nhân vật.
- Sử dụng Animator để đặt trạng thái của nhân vật có đang chạy, nhảy, đứng yên.
- Physics2D.Raycast dùng để kiểm tra nhân vật xem có đang ở trên mặt đất hay không bằng cách sử dụng tia Raycast hướng xuống vị trí hiện tại của nhân vật.
- Nếu người chơi bấm vào phím W và nhân vật đang ở trên mặt đất thì nhân vật sẽ nhảy lên.
- Nếu người chơi bấm vào phím Space và thời điểm hiện tại lớn hơn thời điểm bắn cuối cùng cộng thêm 0.25 giây thì nhân vật sẽ bắn ra đạn, thời điểm bắn cuối cùng sẽ được cập nhật.

```
void Update()
{
    Die();
    Horizontal = Input.GetAxisRaw("Horizontal");
    if (Horizontal < 0.0f) transform.localScale = new Vector3(-1.0f, 1.0f, 1.0f);
    else if (Horizontal > 0.0f) transform.localScale = new Vector3(1.0f, 1.0f, 1.0f);

    Animator.SetBool("running", Horizontal != 0.0f);
    //Debug.DrawRay(transform.position, Vector3.down * 0.1f, color: Color.red);
    if (Physics2D.Raycast(transform.position, Vector3.down, 0.12f))
    {
        Grounded = true;
    }
    else
    {
        Grounded = false;
    }
    if (Input.GetKeyDown(KeyCode.W) && Grounded)
    {
        Jump();
    }

    if (Input.GetKeyDown(KeyCode.Space) && Time.time > LastShoot + 0.25f)
    {
        Shoot();
        LastShoot = Time.time;
    }
}
```



*Hàm Jump()*: Xử lý nhân vật nhảy lên khi có sự kiện bấm vào phím W của người chơi. Cụ thể thì nó sẽ thực hiện thêm một lực vào nhân vật, khiến nhân vật nhảy lên. `Vector2.up` là một vector hướng lên, giá trị `JumpForce` là một biến đại diện cho lực nhảy của nhân vật, độ lớn của lực phụ thuộc vào giá trị này.

*Hàm Shoot()*: Xử lý nhân vật bắn đạn

- Đầu tiên ta phải xác định hướng của đường đạn bay.
- Đặt hướng của đạn dựa trên hướng nhân vật đang quay. Nếu nhân vật đang quay về hướng chiều dương (`scale.x == 1.0f`), viên đạn sẽ chuyển động về hướng phải; ngược lại thì nó sẽ chuyển động về hướng trái.
- Tiếp theo sẽ tạo ra một đối tượng từ prefab viên đạn tại vị trí bên cạnh nhân vật (cách nhân vật một khoảng nhỏ). Sau đó thiết lập đối tượng đạn vừa tạo và đặt vào hướng di chuyển của nó thông qua script `BulletScript`.

```
1 reference
private void Jump()
{
    Rigidbody2D.AddForce(Vector2.up * JumpForce);
}

1 reference
private void Shoot()
{
    Vector3 direction;
    if (transform.localScale.x == 1.0f)
    {
        direction = Vector3.right;
    }
    else
    {
        direction = Vector3.left;
    }
    GameObject bullet = Instantiate(BulletPrefab, transform.position + direction * 0.1f, Quaternion.identity);
    bullet.GetComponent<BulletScript>().SetDirection(direction);

    //Debug.Log($"Shoot Direction: {direction}");
    //Debug.Log($"Shoot Position: {transform.position + direction * 0.1f}");
}
```

*Hàm Hit() và Die()*: Xác định nhân vật nhận sát thương và kiểm tra trạng thái nhân vật còn sống hay đã chết. Cụ thể thì hàm sẽ thực hiện giảm lượng máu của nhân vật đi khi nhân vật bị nhân sát thương đi một lượng bằng với giá trị `damage`. Sau đó lượng máu của nhân vật sẽ được cập nhật trên giao diện. Đối với hàm `Die()` thì sẽ kiểm tra các kiểu chết của nhân vật: hết máu, vị trí nhảy xuống vực, ...

```
public bool IsDie { get { return _isDie; } }  
1 reference  
public void Hit(int damage = 1)  
{  
    _currentHealth -= damage;  
    playerhealth.SetHealth(_currentHealth);  
    Debug.Log("Damage to John");  
    if (_currentHealth == 0)  
    {  
        _isDie = true;  
        Destroy(gameObject);  
    }  
}
```

### File code BulletScript.cs:

*Hàm FixedUpdate():* hàm này được gọi sau mỗi một khung hình vật lý. Trong hàm này vận tốc của Rigidbody2D (thành phần vật lý của viên đạn) được thiết lập bằng giá trị Direction \* Speed, tức viên đạn sẽ di chuyển theo hướng Direction với vận tốc Speed đã được khai báo từ trước.

*Hàm SetDirection():* nhằm gán giá trị Direction được khai báo từ trước.

*Hàm DestroyBulltet():* nhằm hủy đối tượng đang gán script này, cụ thể ở đây là hủy viên đạn này đi.

*Hàm OnTriggerEnter2D():* Phương thức xử lý viên đạn tấn công vào nhân vật hoặc kẻ thù. Cụ thể là gọi đối tượng đã va chạm với viên đạn, nếu khác null thì giảm số máu đi một lượng(được truyền vào hàm). Sau đó hủy đối tượng đạn này đi. Nếu 2 loại đạn bắn vào nhau thì hủy cả 2 đạn.

```

@ Unity Message | 0 references
private void FixedUpdate()
{
    Rigidbody2D.velocity = Direction * Speed;
}

3 references
public void SetDirection(Vector3 direction)
{
    Direction = direction;
}

1 reference
public void DestroyBullet()
{
    Destroy(gameObject);
}

@ Unity Message | 0 references
private void OnTriggerEnter2D(Collider2D other)
{
    JohnMovement john = other.GetComponent<JohnMovement>();
    GruntScript grunt = other.GetComponent<GruntScript>();
    if (grunt != null)
    {
        grunt.Hit(1);
    }
    if (john != null)
    {
        john.Hit(1);
    }
}

```

### File code CameraScript.cs:

Hàm *Update()*: Cập nhật vị trí camera, bám theo nhân vật game.

- Kiểm tra xem nhân vật chính có tồn tại hay không.
- Lấy vị trí hiện tại của Camera, gán cho biến Pos.
- Đặt giá trị x của biến Pos bằng giá trị x của nhân vật game, điều này có nghĩa là camera luôn cùng vị trí của nhân vật chính, tạo ra hiệu ứng theo dõi nhân vật theo chiều ngang.
- Cập nhật vị trí mới của Camera với giá trị mới.

```

}

// Update is called once per frame
@ Unity Message | 0 references
void Update()
{
    if (John != null)
    {
        Vector3 pos = transform.position;
        pos.x = John.transform.position.x;

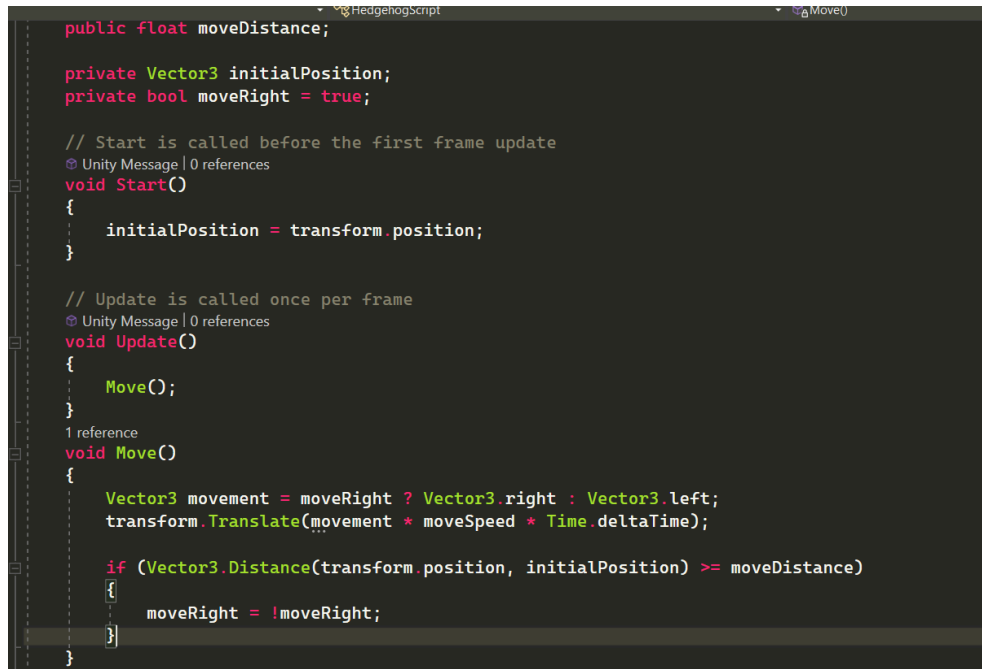
        transform.position = pos;
    }
}

```

### File code HedgehogScript.cs:

Hàm *Move()*: di chuyển đối tượng trong không gian.

- Đối tượng tự động di chuyển sang trái, sang phải, nếu đến một ngưỡng move Distance nào đó cho trước thì chiều di chuyển moveRight sẽ bị đảo ngược.



```
public float moveDistance;

private Vector3 initialPosition;
private bool moveRight = true;

// Start is called before the first frame update
@ Unity Message | 0 references
void Start()
{
    initialPosition = transform.position;
}

// Update is called once per frame
@ Unity Message | 0 references
void Update()
{
    Move();
}

1 reference
void Move()
{
    Vector3 movement = moveRight ? Vector3.right : Vector3.left;
    transform.Translate(movement * moveSpeed * Time.deltaTime);

    if (Vector3.Distance(transform.position, initialPosition) >= moveDistance)
    {
        moveRight = !moveRight;
    }
}
```

Ngoài ra các file code khác chứa các phương thức bắt sự kiện button để lựa chọn level, trở lại, giao diện HUD để hiển thị máu, score, ...

---

## TÀI LIỆU THAM KHẢO

- [1] *Tìm hiểu về Unreal Engine*. (2023, May 18). Ganbaru. Retrieved November 9, 2023, from <https://gambaru.io/en/blog/tim-hieu-ve-unreal-engine>
- [2] *Unity là gì? Các lỗi thường gặp khi làm việc với Unity là gì?* (n.d.). ITdev. Retrieved November 9, 2023, from <https://itviec.com/blog/unity-la-gi/>
- [3] “”Giải mã” quy trình phát triển ứng dụng Game hiệu quả” FPT Polytechnic. Retrieved December 9, 2023, from <https://caodang.fpt.edu.vn/tin-tuc-poly/tin-da-nang/giai-ma-quy-trinh-phat-trien-ung-dung-game-hieu-qua.html>
- [4] “Quy trình sản xuất Game và những vị trí quan trọng trong mảng game Art”. MAAC, Retrieved December 9, 2023, from <https://maac.edu.vn/vi/quy-trinh-san-xuat-game-va-nhung-vi-tri-quan-trong-trong-mang-game-art>
- [5] “Tìm hiểu về mô hình MVVM.” Viblo, Accessed 10 January 2024. from <https://viblo.asia/p/tim-hieu-ve-mo-hinh-mvvm-maGK7vW95j2>.