

## NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

*Vĩnh Long, ngày ... tháng ... năm 2025*

**Giáo viên hướng dẫn**

*(Ký và ghi rõ họ tên)*

## NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

*Vĩnh Long, ngày ... tháng ... năm 2025*

**Thành viên hội đồng**

*(Ký và ghi rõ họ tên)*

## LỜI CẢM ƠN

Đầu tiên, em xin gửi lời cảm ơn chân thành đến các thầy cô, những người đã hết mực chỉ bảo, hướng dẫn và sẵn sàng giải đáp thắc mắc, cung cấp những kiến thức hữu ích giúp em hoàn thành công việc trong suốt quá trình thực hiện báo cáo này.

Mặc dù đã cố gắng để thực hiện đề tài một cách nghiêm túc và tập trung trong suốt quá trình thu thập, tìm hiểu, phân tích và tổng hợp nhiều nguồn tài liệu khác nhau, song do sự hiểu biết còn chưa cao, trình độ chuyên môn còn hạn chế ở nhiều mặt và chưa có nhiều kỹ năng trong việc thực hiện các công việc trong chuyên ngành nên trong quá trình biên soạn đồ án không thể tránh khỏi các thiết sót. Do đó nếu như có những điểm trong báo cáo còn chưa hợp lý hoặc sai sót em rất mong được các thầy cô đóng góp thêm các ý kiến để có thể học hỏi và tiếp thu thêm các kiến thức còn thiếu sót.

Em xin chân thành cảm ơn!

Sinh viên thực hiện

**Ngô Huỳnh Quốc Khang**

## Mục Lục

CHƯƠNG 1. TỔNG QUAN .....	1
1.1. Giới thiệu đề tài .....	1
1.2. Mục tiêu của đề tài .....	1
1.3. Đối tượng nghiên cứu.....	1
1.4. Phạm vi nghiên cứu .....	2
1.5. Phương pháp nghiên cứu .....	2
1.5.1. Phương pháp phân tích và thiết kế .....	2
1.5.2. Phương pháp xây dựng hệ thống.....	2
1.5.3. Phương pháp kiểm thử.....	3
1.5.4. Phương pháp triển khai.....	3
1.5.5. Phương pháp khảo sát và đánh giá .....	3
CHƯƠNG 2. NGHIÊN CỨU LÝ THIẾT .....	4
2.1. Tổng quan về game thể loại nhập vai.....	4
2.2. Tổng quan về đồ họa pixel .....	4
2.3. Tổng quan về Godot Engine.....	5
2.4. Tổng quan về Krita.....	6
2.5. Tổng quan về FL Studio .....	6
2.6. Tổng quan về JSON.....	6
CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU.....	7
3.1. Mô tả bài toán .....	7

3.2. Phân tích và thiết kế hệ thống.....	7
3.2.1. Đặc tả yêu cầu hệ thống.....	7
3.2.2. Tổ chức lưu trữ dữ liệu.....	9
3.2.3. Thiết kế xử lý.....	17
3.2.4. Thiết kế giao diện.....	20
3.2.5. Thiết kế asset.....	23
CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU .....	30
4.1. Dữ liệu thử nghiệm.....	30
4.1.1. Dữ liệu đối thoại thử nghiệm (NPC Dialog JSON).....	30
4.1.2. Dữ liệu thử nghiệm nhiệm vụ (Quest Resource).....	31
4.1.3. Dữ liệu vật phẩm thử nghiệm (Inventory Item) .....	32
4.2. Kết quả thực nghiệm.....	33
4.2.1. Kết quả kiểm thử di chuyển, va chạm và camera.....	33
4.2.2. Kết quả kiểm thử hệ thống đối thoại NPC .....	33
4.2.3. Kết quả kiểm thử hệ thống nhiệm vụ (Quest) .....	34
4.2.4. Kết quả kiểm thử hệ thống vật phẩm và túi đồ (Inventory) .....	34
4.2.5. Kết quả kiểm thử chiến đấu cơ bản .....	34
4.2.6. Kết quả kiểm thử âm thanh và nhạc nền .....	34
4.2.7. Kết quả kiểm thử lưu và tải tiến trình.....	34
4.3. Giao diện và gameplay .....	35
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	48
5.1. Kết luận.....	48

---

5.2. Hạn chế .....	50
5.2.1. Gameplay còn đơn giản .....	50
5.2.2. Thiếu chiều sâu nội dung .....	50
5.2.3. Hạn chế về đồ họa và hoạt ảnh .....	51
5.2.4. Âm thanh còn tối giản .....	51
5.2.5. AI NPC và quái vật đơn giản .....	51
5.2.6. Hệ thống lưu trữ dữ liệu còn thủ công .....	51
5.3. Hướng phát triển .....	52
5.3.1. Mở rộng gameplay và hệ thống chiến đấu .....	52
5.3.2. Hoàn thiện hệ thống nhiệm vụ .....	52
5.3.3. Mở rộng thế giới game .....	53
5.3.4. Cải thiện đồ họa và hiệu ứng .....	53
5.3.5. Tối ưu UI và trải nghiệm người chơi .....	53
5.3.6. Nâng cấp hệ thống lưu trữ .....	54
5.3.7. Xuất bản đa nền tảng .....	54
TÀI LIỆU THAM KHẢO .....	55

## **DANH MỤC HÌNH ẢNH – BẢNG BIỂU**

Hình 3.1 Sơ đồ Use Case của Actor Player .....	17
Hình 3.2 Sơ đồ User Case của Actor NPC .....	18

Hình 3.3 Sơ đồ Use Case của Actor Quái vật.....	19
Hình 3.4 Phác thảo giao diện màn hình chính.....	20
Hình 3.5 Phác thảo giao diện màn hình tạm dừng game .....	20
Hình 3.6 Phác thảo giao diện túi đồ .....	21
Hình 3.7 Phác thảo giao diện hội thoại .....	22
Hình 3.8 Phác thảo giao diện danh sách nhiệm vụ .....	22
Hình 3.9 Hoạt ảnh chuyển động của Player .....	23
Hình 3.10 Hoạt ảnh chuyển động của kẻ địch.....	23
Hình 3.11 Hoạt ảnh chuyển động của NPC hội trường.....	24
Hình 3.12 Hoạt ảnh chuyển động của trưởng làng.....	25
Hình 3.13 Hoạt ảnh chuyển động của người bán hàng .....	26
Hình 3.14 Hoạt ảnh chuyển động của lính canh.....	27
Hình 3.15 Tileset xây dựng môi trường.....	28
Hình 3.16 Các khung để xây dựng giao diện hội thoại và túi đồ .....	29
Hình 3.17 Vật phẩm.....	29
Hình 3.18 Kiếm .....	29
Hình 3.19 Hiệu ứng nổ .....	29
Hình 4.1 Giao diện trang chủ .....	35
Hình 4.2 Giao diện gameplay của game.....	36
Hình 4.3 Player di chuyển sang map khác .....	36
Hình 4.4 Giao diện tạm dừng game.....	37
Hình 4.5 Giao diện hộp thoại giao tiếp giữa player và NPC.....	38

---

Hình 4.6 Đoạn hội thoại để player nhận nhiệm vụ.....	38
Hình 4.7 Player nhận nhiệm vụ thành công .....	39
Hình 4.8 Giao diện danh sách nhiệm vụ.....	39
Hình 4.9 Player tấn công kẻ địch.....	40
Hình 4.10 Player nhận sát thương từ kẻ địch khi bị đánh trúng.....	41
Hình 4.11 Kẻ địch rơi ra vật phẩm khi bị hạ gục .....	42
Hình 4.12 Khi người chơi đến gần vật phẩm, giao diện sẽ chỉ dẫn cách nhặt vật phẩm ...	42
Hình 4.13 Khi người chơi nhặt đúng vật phẩm cần thiết cho nhiệm vụ thì tiến trình trong giao diện nhiệm vụ sẽ tăng lên .....	43
Hình 4.14 Khi hoàn thành nhiệm vụ thì người chơi sẽ nhận được 3 coin cộng vào ở góc trên bên phải màn hình .....	44
Hình 4.15 Player giao tiếp với người bán hàng.....	45
Hình 4.16 Player mua vật phẩm thành công .....	45
Hình 4.17 Giao diện túi đồ của Player .....	46
Hình 4.18 Giao diện hiển thị thông tin chi tiết về vật phẩm .....	46
Hình 4.19 Giao diện lựa chọn sử dụng hoặc bỏ vật phẩm .....	47
Hình 4.20 Player hồi máu thành công từ 4.0 sang 5.0.....	47
Bảng 3.1 Cấu trúc dữ liệu Resource Quest.....	12
Bảng 3.2 Cấu trúc dữ liệu Resource Objective .....	13
Bảng 3.3 Cấu trúc dữ liệu Resource Reward .....	14
Bảng 3.4 Cấu trúc dữ liệu lưu trữ dữ liệu vật phẩm.....	16

---



Bảng 4.1 Bảng dữ liệu dialog .....	30
Bảng 4.2 Bảng dữ liệu nhiệm vụ .....	32
Bảng 4.3 Bảng dữ liệu mục tiêu của nhiệm vụ.....	32
Bảng 4.4 Bảng dữ liệu phần thưởng của nhiệm vụ .....	32
Bảng 4.5 Bảng dữ liệu vật phẩm .....	33

## CHƯƠNG 1. TỔNG QUAN

### 1.1. Giới thiệu đề tài

Đề tài tập trung vào việc nghiên cứu các phương pháp nhằm xây dựng một tựa game nhập vai 2D với góc nhìn từ trên xuống (top-down). Tựa game được thiết kế theo phong cách đồ họa pixel cho phép người chơi điều khiển nhân vật chính di chuyển trong một ngôi làng, tương tác với các nhân vật NPC, khám phá hầm mộ, chiến đấu với quái vật, thu thập tài nguyên để trao đổi, mua vật phẩm và trang bị.

### 1.2. Mục tiêu của đề tài

Mục tiêu chính của đề tài là xây dựng một trò chơi 2D hoàn chỉnh ở mức cơ bản, tập trung vào các thành phần cốt lõi trong phát triển game bằng Godot Engine. Game tập trung vào xây dựng các thành phần chính bao gồm:

- Thiết kế các tileset và hoạt ảnh chuyển động.
- Thiết kế giao diện người dùng (UI) thân thiện.
- Tạo và xử lý âm thanh, nhạc nền.
- Xây dựng hệ thống di chuyển, va chạm và camera theo dõi nhân vật.
- Xây dựng hệ thống quản lý bản đồ, quản lý vật phẩm, túi đồ, mua bán vật phẩm.
- Xây dựng hệ thống chiến đấu cơ bản.
- Xây dựng hệ thống lưu và tải dữ liệu người chơi.

### 1.3. Đối tượng nghiên cứu

Đặc điểm của các game thuộc thể loại nhập vai. Phong cách đồ họa của game dạng 2D, pixel và góc nhìn top - down.

Tìm hiểu tổng quan về Godot Engine, ngôn ngữ GDScript, cấu hình độ phân giải, Scene/Node, TileMap, Signal và Group.

Tìm hiểu cách sử dụng công cụ Krita để thiết kế Sprite, Animation và Tileset theo phong cách pixel.

Tìm hiểu phần mềm FL Studio để tạo và xử lý âm thanh phục vụ cho việc tăng tính trải nghiệm tựa game.

## **1.4. Phạm vi nghiên cứu**

Game không xây dựng cốt truyện phức tạp mà chỉ sử dụng bối cảnh đơn giản và tập trung vào xây dựng cơ chế gameplay và thiết kế đồ họa và âm thanh cho game. Game diễn ra trong hai khu vực chính là:

- Làng: nơi người chơi có thể di chuyển, tương tác với NPC, mua bán vật phẩm và chuẩn bị trước khi vào hầm mỏ.
- Hầm mỏ: khu vực người chơi sẽ vào chiến đấu với các quái vật và thu thập vật phẩm để hoàn thành các yêu cầu của nhiệm vụ.

## **1.5. Phương pháp nghiên cứu**

### **1.5.1. Phương pháp phân tích và thiết kế**

Phân tích và lên ý tưởng gameplay, cơ chế di chuyển, chiến đấu, bản đồ.

Thiết kế đồ họa cho nhân vật, môi trường, vật phẩm, animation và giao diện.

Tạo nhạc nền và hiệu ứng âm thanh bằng FL Studio.

Thiết kế cấu trúc Scene/Node Trong Godot Engine.

### **1.5.2. Phương pháp xây dựng hệ thống**

Xây dựng cấu trúc Scene/Node cho từng phần (nhân vật, bản đồ, UI). Lập trình điều khiển nhân vật, va chạm, AI quái vật, hệ thống chiến đấu và thu thập vật phẩm bằng GDScript. Tạo hệ thống lưu tiến trình (save/load), giao diện menu và quản lý dữ liệu bằng file JSON và Dictionary trong Godot.

### **1.5.3. Phương pháp kiểm thử**

Kiểm tra từng phần của game sau khi hoàn thành để phát hiện lỗi.

Chơi thử toàn bộ game nhiều lần để đánh chất lượng, độ khó và trải nghiệm.

Sửa lại những phần chưa ổn. Đóng gói và triển khai.

### **1.5.4. Phương pháp triển khai**

Xuất file chạy game trên máy tính (.exe) bằng công cụ xuất của Godot.

Kiểm tra lại sau khi xuất file để đảm bảo game vẫn hoạt động bình thường.

Chuẩn bị video demo và hình ảnh minh họa để phục vụ cho việc báo cáo.

### **1.5.5. Phương pháp khảo sát và đánh giá**

Thu thập phản hồi từ bạn bè, giảng viên hoặc người chơi thử.

Dựa vào ý kiến đánh giá để cải tiến gameplay, chỉnh sửa giao diện, cân bằng độ khó.

Tổng hợp kết quả đạt được, rút ra kinh nghiệm và hướng phát triển game tiếp theo.

## CHƯƠNG 2. NGHIÊN CỨU LÝ THIẾT

### 2.1. Tổng quan về game thể loại nhập vai

Game nhập vai hay còn gọi là game RPG (Role-Playing Game), là một thể loại game mà trong đó người chơi sẽ được hóa thân vào một hoặc nhiều nhân vật, tham gia vào những cuộc phiêu lưu, khám phá thế giới trong game và phát triển kỹ năng nhân vật theo thời gian.

Điểm đặc trưng của thể loại này là nằm ở phần cốt truyện được xây dựng rất chi tiết, nhiều phân nhánh khác nhau và có thể thay đổi dựa trên quyết định của người chơi, đồng thời hệ thống NPC trong game cũng được xây dựng chi tiết với các tính cách khác nhau kèm theo câu truyện của riêng của từng NPC.

Người chơi sẽ trải qua nhiều nhiệm vụ chính và phụ khác nhau, trong đó tùy vào các quyết định của người chơi ở phần cốt truyện sẽ dẫn đến sự thay đổi về phần nhiệm vụ chính nhưng không làm thay phần nhiệm vụ phụ.

Trong bất kỳ tựa game nhập vai nào, hệ thống chiến đấu sẽ luôn giữ vai trò cốt lõi, định hình trải nghiệm và cảm xúc của người chơi. Hệ thống này thường được chia thành hai dạng là chiến đấu theo lượt có nhịp độ rất chậm và mang tính chiến thuật cao và thời gian suy nghĩ lâu, buộc người chơi phải cân nhắc kỹ lưỡng từng hành động trước khi ra nước đi tiếp theo. Ngược lại với đó là chiến đấu thời gian thực có nhịp độ mang tính rất nhanh và dồn dập, người chơi phải dựa vào phản xạ nhanh để dành chiến thắng, từ đó tạo ra sự kịch tính và căng thẳng.

### 2.2. Tổng quan về đồ họa pixel

Đồ họa pixel là kỹ thuật sử dụng các điểm ảnh làm đơn vị cơ bản để tạo hình ảnh, thường được sử dụng trong các game 2D cổ điển và game indie hiện nay.

Đặc điểm nổi bật của đồ họa pixel là khả năng truyền tải hình ảnh chi tiết với lượng dữ liệu cực nhỏ, tạo nên cảm giác cổ điển và gợi nhớ những game đồ họa 8-bit và 16-bit của những năm 1980 và 1990.

Kỹ thuật này cho phép nhà phát triển dễ dàng tùy chỉnh từng frame của animation, tạo các sprite cho nhân vật, quái vật, vật phẩm, cũng như các tileset để xây dựng môi trường game như nền đất, cây cối hay tường.

Pixel art cũng thường được sử dụng để làm icon, hiệu ứng đơn giản như cháy nổ hay ánh sáng, mang lại sự linh hoạt và tiết kiệm tài nguyên cho game.

### 2.3. Tổng quan về Godot Engine

Godot Engine là một công cụ phát triển game đa nền tảng với nhiều tính năng nổi bật, cho phép tạo ra các trò chơi 2D và 3D từ một giao diện thống nhất. Nó cung cấp một bộ công cụ toàn diện và phổ biến, giúp người dùng tập trung vào việc làm game mà không cần phải tự tạo lại từ đầu. Game có thể được xuất ra với một cú nhấp chuột sang nhiều nền tảng, bao gồm các nền tảng máy tính để bàn chính (Linux, macOS, Windows), nền tảng di động (Android, iOS), cũng như các nền tảng web và máy chơi game [1].

Godot Engine hoàn toàn miễn phí và mã nguồn mở theo giấy phép MIT tự do. Không ràng buộc, không phí bản quyền, không gì cả. Trò chơi của người dùng thuộc về họ, đến từng dòng mã cuối cùng của engine. Quá trình phát triển của Godot hoàn toàn độc lập và do cộng đồng dẫn dắt, trao quyền cho người dùng tham gia định hình engine sao cho phù hợp với kỳ vọng của họ. Nó được hỗ trợ bởi Quỹ Godot, một tổ chức phi lợi nhuận [1].

Godot Engine đặc biệt mạnh ở mảng 2D với các công cụ tilemap, sprite animation, camera và physics 2D và signal nhưng vẫn hỗ trợ cơ bản đến nâng cao cho 3D với mesh, light, material và animation. Godot trở thành lựa chọn lý tưởng cho các dự án game indie vì có mã nguồn mở, dung lượng nhẹ, giao diện linh hoạt dễ sử dụng, rất thích hợp cho các nhà phát triển có nhu cầu cần nhanh chóng thử nghiệm ý tưởng. Godot còn tích hợp sẵn các công cụ để đọc và ghi dữ liệu từ JSON, cho phép lưu trữ thông tin nhân vật, level hay item một cách linh hoạt.

## 2.4. Tổng quan về Krita

Krita là phần mềm vẽ kỹ thuật số miễn phí, nổi bật với khả năng tạo tranh, sprite và animation 2D. Krita cung cấp nhiều công cụ mạnh mẽ như layer và group layer giúp quản lý sprite và background một cách chi tiết, cùng với hệ thống brush đa dạng hỗ trợ vẽ texture và các chi tiết nhỏ. Ngoài ra, Krita còn hỗ trợ animation frame, cho phép tạo các sprite sheet hoặc các animation ngắn trực tiếp trong phần mềm, đồng thời hỗ trợ xuất file dưới nhiều định dạng phổ biến như PNG, PSD hay GIF. Trong phát triển game, Krita thường được sử dụng để tạo sprite nhân vật, tileset môi trường, background và UI.

## 2.5. Tổng quan về FL Studio

FL Studio là một phần mềm sản xuất nhạc kỹ thuật số phổ biến nhất hiện nay, được sử dụng để tạo nhạc nền và hiệu ứng âm thanh. FL Studio cung cấp nhiều công cụ mạnh mẽ, từ piano roll để ghi chú nhạc, thiết lập melody, chord progression, đến mixer và các hiệu ứng như reverb, EQ, compression, giúp âm thanh trở nên sống động và chuyên nghiệp.

FL Studio có khả năng xuất file dưới nhiều định dạng như WAV, MP3 hay MIDI, giúp dễ dàng tích hợp vào dự án game, từ đó nâng cao trải nghiệm người chơi thông qua âm thanh hiệu ứng và nhạc nền sống động.

## 2.6. Tổng quan về JSON

JSON (JavaScript Object Notation) là một định dạng dữ liệu nhẹ, dễ đọc, dễ ghi, thường được sử dụng để lưu trữ và trao đổi thông tin qua nhiều nền tảng khác nhau.

JSON có thể lưu trữ dữ liệu dưới dạng object hoặc array, rất phù hợp để quản lý trạng thái game, thuộc tính nhân vật, item, level hay skill.

## CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU

### 3.1. Mô tả bài toán

Đề tài tập trung xây dựng một trò chơi nhập vai 2D theo phong cách pixel art với góc nhìn từ trên xuống (top-down). Người chơi điều khiển một nhân vật di chuyển trong môi trường gồm khu vực làng và hầm mỏ, thực hiện tương tác cơ bản, thu thập vật phẩm và tham gia chiến đấu. Game tập trung vào yếu tố triển khai các cơ chế gameplay.

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

#### 3.2.1. Đặc tả yêu cầu hệ thống

##### 3.2.1.1. Yêu cầu chức năng

Hệ thống cần đảm bảo nhân vật có thể di chuyển tự do trong thế giới game theo bốn hướng cơ bản là trái, phải, lên và xuống với animation tương ứng cho từng hướng để phản ánh đúng chiều chuyển động. Việc di chuyển không chỉ là thay đổi vị trí mà còn phải gắn liền với việc chạy hoạt ảnh mềm mại, nhất quán và trực quan, giúp người chơi cảm thấy nhân vật sống động và phản ứng đúng với thao tác đầu vào. Bên cạnh chuyển động, nhân vật phải có khả năng tương tác với môi trường, chẳng hạn như nói chuyện với NPC, mở túi đồ, nhặt vật phẩm, kích hoạt cơ chế hay chuyển map, tất cả đều dựa trên hệ thống phát hiện tương tác hợp lý.

Hệ thống va chạm đóng vai trò then chốt trong trải nghiệm gameplay. Va chạm phải được xử lý chuẩn xác giữa nhân vật, quái vật, cũng như các tile trên bản đồ, đảm bảo rằng nhân vật hoặc quái không đi xuyên tường hoặc lọt xuống nền và mọi đối tượng đều có vùng va chạm được kiểm soát chặt chẽ. Việc quản lý va chạm phải giúp tất cả thực thể trong game thể hiện đúng hành vi mong muốn, không gây cảm giác bất thường hoặc lỗi hình học.

Đối với quái vật, yêu cầu chức năng tập trung vào hành vi đơn giản nhưng rõ ràng, quái có thể di chuyển trong phạm vi của chúng, phát hiện người chơi dựa trên điều kiện đặt



trước (khoảng cách hoặc tầm nhìn) và tấn công khi tiếp cận mục tiêu. Hành vi quái phải đủ nhất quán để truyền tải cảm giác tương tác thật sự với môi trường sống động.

Người chơi ngoài di chuyển còn có cơ chế chiến đấu, bao gồm khả năng tấn công, nhận sát thương và sử dụng vật phẩm. Hệ thống chiến đấu phải phản hồi ngay lập tức với thao tác người chơi, hiển thị rõ khi gây sát thương hoặc nhận sát thương, đồng thời cho phép sử dụng vật phẩm để hỗ trợ chiến đấu. Mọi hành động liên quan đến chiến đấu và vật phẩm phải hoạt động thống nhất với hệ thống animation và va chạm.

Hệ thống túi đồ, cho phép người chơi lưu trữ, sử dụng và trao đổi, nhặt hoặc bỏ đồ. Túi đồ phải có khả năng quản lý vật phẩm theo số lượng, phân loại và hiển thị rõ ràng, giúp người chơi dễ dàng chuyển đổi giữa việc dùng đồ trong chiến đấu, lưu lại đồ cho sau này hoặc bán cho NPC.

Hệ thống túi đồ liên kết chặt chẽ với hệ thống cửa hàng, nơi người chơi có thể mua vật phẩm với NPC. Cửa hàng phải phản hồi rõ ràng khi người chơi mua hoặc bán vật phẩm, tự động cập nhật số lượng trong túi đồ của người chơi, giá trị giao dịch và thông báo phù hợp khi người chơi trao đổi đồ với NPC.

Ngoài ra, hệ thống map phải cho phép người chơi chuyển đổi giữa làng và hầm mỏ, tạo ra sự đa dạng về môi trường và gameplay. Việc chuyển map cần mượt mà, chính xác, không gây giật lag hoặc lỗi vị trí nhân vật. Khi di chuyển giữa làng và hầm mỏ, trạng thái của người chơi như máu, vật phẩm, quái hiện diện phải được xử lý ổn định.

Cuối cùng, toàn bộ dữ liệu của game, bao gồm trạng thái nhân vật, túi đồ, vị trí map và các tiến trình nhiệm vụ khi chơi, cần được lưu và tải bằng JSON, đảm bảo dữ liệu dễ đọc, dễ chỉnh sửa, dễ mở rộng về sau. JSON giúp đảm bảo tính tương thích, nhẹ và phù hợp cho nhiều nền tảng.

### 3.2.1.2. Yêu cầu phi chức năng

Game cần sở hữu giao diện trực quan và dễ thao tác, mọi nút bấm, menu, biểu tượng đều phải được bố trí hợp lý, dễ hiểu ngay từ lần đầu nhìn thấy. Giao diện không gây rối mắt và giúp người chơi thao tác nhanh chóng, từ việc điều khiển nhân vật, mở túi đồ, mua bán vật phẩm đến di chuyển giữa các màn chơi.

Về mặt hình ảnh, game sử dụng phong cách pixel art, vì vậy hình ảnh phải rõ nét, không được vỡ hay co kéo sai tỉ lệ. Mọi sprite, animation, tile map phải được hiển thị ở đúng độ phân giải dự kiến, không bị giãn méo khi phóng to hoặc thu nhỏ. Chất lượng pixel phải được giữ ổn định trong mọi tình huống như di chuyển camera, phóng to màn hình hoặc chuyển giữa các map.

Game phải đạt FPS ổn định trên cấu hình máy tính phổ thông, nghĩa là không cần máy mạnh vẫn chơi mượt. FPS phải giữ ở mức cao và ổn định dù có nhiều quái, nhiều hiệu ứng, hay map lớn, nhằm đảm bảo trải nghiệm liền mạch, không giật lag.

Về mặt tổ chức mã nguồn, code cần dễ mở rộng và được sắp xếp rõ ràng theo Scene Node. Mỗi thành phần trong game như nhân vật, quái, UI, hệ thống túi đồ, hệ thống cửa hàng... phải nằm trong cấu trúc tổ chức hợp lý, dễ theo dõi, dễ chỉnh sửa và dễ thêm chức năng mới. Điều này giúp việc phát triển lâu dài, bảo trì hoặc cập nhật trở nên thuận tiện.

### 3.2.2. Tổ chức lưu trữ dữ liệu

Hệ thống lưu trữ dữ liệu trong game được xây dựng theo hướng linh hoạt và dễ mở rộng, kết hợp giữa file JSON cho dữ liệu động hoặc dữ liệu có khả năng thay đổi theo nội dung game và Resource (.tres/.res) của Godot để lưu các cấu trúc dữ liệu có tính ổn định cao, có kiểu dữ liệu rõ ràng và được hỗ trợ trực tiếp bởi Editor.

Việc sử dụng song song hai cơ chế này giúp game vừa dễ quản lý dữ liệu có cấu trúc nội dung cao như dialog, quest và item, vừa đảm bảo hiệu năng, độ an toàn và độ tin cậy khi try cập, đọc và ghi dữ liệu gameplay của toàn bộ tựa game.

### 3.2.2.1. Lưu trữ hội thoại NPC bằng JSON

Hệ thống hội thoại sử dụng một file JSON duy nhất để chứa toàn bộ cây hội thoại (dialog trees) của các NPC trong game. Cách tổ chức này rất phù hợp cho game RPG vì:

- Người thiết kế chỉ cần chỉnh sửa file JSON mà không cần đụng đến code.
- Mỗi NPC có thể sở hữu nhiều nhánh hội thoại (branch), dễ mở rộng.
- Mỗi dialog được mô tả thống nhất bằng các trường cơ bản: state, text, options.
- Cấu trúc dạng tree giúp mô phỏng hội thoại dạng phân nhánh tương tự các game 2D RPG cổ điển.

#### 3.2.2.1.1 Cấu trúc dữ liệu mẫu

Cấu trúc dữ liệu mẫu dưới đây minh họa cách hệ thống lưu trữ và tổ chức thông tin hội thoại của một nhân vật trong game, bao gồm các nhánh hội thoại, trạng thái và lựa chọn mà người chơi có thể tương tác.

```
"trưởng làng": {
  "trees": [
    {
      "branch_id": "chào hỏi",
      "dialogs": [
        {
          "state": "start",
          "text": "Xin chào.",
          "options": { "Rời đi": "exit" }
        },
        {
          "state": "exit",
          "text": "",
          "options": {}
        }
      ]
    }
  ]
}
```

### 3.2.2.1.2 Phân tích cấu trúc

NPC Name (key cấp 1): mỗi NPC có một tên duy nhất trong JSON.

trees: danh sách các nhánh hội thoại. Mỗi nhánh tương ứng một tình huống hoặc trạng thái khác nhau của NPC (ví dụ: “chào hỏi”, “nhận nhiệm vụ”, “mua hàng”).

branch\_id: định danh để hệ thống gọi đúng nhánh phù hợp.

dialogs: danh sách các state hội thoại. Mỗi state tuân theo mô hình:

- state: tên trạng thái
- text: câu thoại
- options: các lựa chọn của người chơi

### 3.2.2.2. Hệ thống nhiệm vụ (Quest) dùng Resource

Trong hệ thống nhiệm vụ, toàn bộ dữ liệu đều được lưu trữ bằng Resource. Đây là một cách làm rất phổ biến và cũng cực kỳ đơn giản khi làm việc với Godot, vì Resource mang lại nhiều lợi ích rõ ràng.

Resource có khả năng tự động ghi và đọc dữ liệu (serialize/deserialize). Điều này có nghĩa là người dùng không cần tự viết các đoạn mã phức tạp để chuyển dữ liệu sang dạng có thể lưu vào file rồi lại chuyển ngược trở lại khi cần sử dụng. Godot sẽ tự xử lý giúp giảm đáng kể thời gian và công sức.

Ngoài ra, Resource còn hỗ trợ export property, cho phép chỉnh sửa các giá trị ngay trong Inspector của Godot. Điều này rất tiện lợi vì người dùng có thể thay đổi nội dung dữ liệu trực quan, không cần mở file hoặc sửa bằng mã nguồn.

Một điểm mạnh nữa là Resource có kiểu dữ liệu rõ ràng như String (chuỗi ký tự), int (số nguyên), Array (mảng), cùng các kiểu dữ liệu nguyên thủy khác. Việc có kiểu dữ liệu cố định giúp giảm thiểu lỗi trong quá trình chạy chương trình, vì Godot có thể kiểm tra và cảnh báo khi người dùng nhập dữ liệu sai.

Resource còn hỗ trợ kế thừa (inheritance), giúp tạo ra nhiều loại dữ liệu khác nhau dựa trên một mẫu có sẵn. Điều này làm cho hệ thống dễ mở rộng, dễ tái sử dụng và tránh việc lặp lại mã hoặc cấu trúc dữ liệu không cần thiết.

**Bảng 3.1 Cấu trúc dữ liệu Resource Quest**

Thuộc tính	Kiểu dữ liệu	Ý nghĩa / Vai trò
quest_id	String	Mã định danh duy nhất của nhiệm vụ. Dùng để hệ thống phân biệt từng Quest và lưu tiến trình người chơi.
quest_name	String	Tên của nhiệm vụ. Hiển thị trong UI như bảng nhiệm vụ hoặc thông báo.
description	String	Mô tả nội dung tổng quan của nhiệm vụ, giúp người chơi hiểu mục tiêu chính họ cần thực hiện.
npc_id	String	ID của NPC giao nhiệm vụ. Hệ thống dùng trường này để kích hoạt UI nhiệm vụ khi người chơi tương tác NPC đó.
dialog	String	Câu thoại hoặc đoạn giới thiệu khi nhận nhiệm vụ. Giúp tăng tính nhập vai và gắn kết giữa NPC – người chơi.
objectives	Array<Object>	Danh sách các mục tiêu thuộc nhiệm vụ. Mỗi phần tử là một Objective (theo bảng Objective bên trên).
rewards	Array<Object>	Danh sách phần thưởng mà người chơi nhận khi hoàn thành nhiệm vụ. Mỗi phần tử là một Reward.

is_completed	bool	Trạng thái hoàn thành nhiệm vụ. Trở thành true khi tất cả Objective trong nhiệm vụ đều đã hoàn thành.
--------------	------	---

**Bảng 3.2 Cấu trúc dữ liệu Resource Objective**

Thuộc tính	Kiểu dữ liệu	Ý nghĩa / Vai trò
id	String	Mã định danh duy nhất cho từng mục tiêu trong nhiệm vụ. Dùng để hệ thống xác định đúng mục tiêu cần cập nhật.
description	String	Mô tả ngắn gọn nội dung mục tiêu, ví dụ: “Thu thập 3 cục than”, “Nói chuyện với trưởng làng”.
target_id	String	ID đối tượng mà mục tiêu liên quan tới. Có thể là tên vật phẩm (ví dụ “coal”) hoặc tên NPC trong nhiệm vụ dạng “talk_to”.
target_type	String	Xác định loại mục tiêu. Các loại hiện tại gồm: "Collection" (thu thập vật phẩm) và "talk_to" (tương tác NPC).
objective_dialog	String	Câu thoại hoặc thông báo hiển thị khi người chơi hoàn thành mục tiêu này. Không bắt buộc, nhưng giúp tăng trải nghiệm.
required_quantity	int	Số lượng vật phẩm cần thu thập đối với mục tiêu dạng Collection.

collected_quantity	int	Số lượng vật phẩm mà người chơi đã thu thập. Hệ thống tự động tăng khi nhặt vật phẩm tương ứng.
is_completed	bool	Trạng thái hoàn thành mục tiêu. Trở thành true khi điều kiện đạt ngưỡng (đủ số lượng hoặc đã nói chuyện NPC).

**Bảng 3.3 Cấu trúc dữ liệu Resource Reward**

Thuộc tính	Kiểu dữ liệu	Ý nghĩa / Vai trò
reward_type	String	Loại phần thưởng người chơi nhận được. Có thể gồm: "gold", "item", "exp", "stat_buff"... tùy cách mở rộng của game.
reward_amount	int	Giá trị phần thưởng tương ứng. Ví dụ: 50 vàng, 1 vật phẩm, 10 điểm kinh nghiệm...

### 3.2.2.3. Hệ thống lưu trữ và tương tác vật phẩm

Hệ thống vật phẩm trong game được thiết kế theo một cấu trúc phân loại rõ ràng, trong đó mỗi loại vật phẩm sẽ được xử lý theo một cơ chế riêng biệt tùy thuộc vào vai trò và mục đích sử dụng của nó trong gameplay. Hệ thống vật phẩm trong game được chia và xử lý thông qua hai hướng khác nhau:

- Vật phẩm liên quan đến nhiệm vụ (quest items): Những vật phẩm này không được đưa vào túi đồ của người chơi. Chúng được xử lý riêng vì thường chỉ phục

vụ cho các bước trong nhiệm vụ, ví dụ như thu thập một món đồ đặc biệt hoặc kích hoạt một sự kiện trong cốt truyện.

- Vật phẩm thông thường: Những vật phẩm này sẽ được thêm trực tiếp vào inventory (túi đồ) của người chơi. Đây có thể là item tiêu hao, vũ khí, nguyên liệu hoặc các loại đồ dùng thông thường mà người chơi có thể sử dụng, bán hoặc quản lý trong túi đồ.

Cách phân tách rõ ràng giữa hai loại vật phẩm này thực sự rất hợp lý trong thể loại game RPG, tránh trường hợp người chơi vô tình làm mất quest item, chẳng hạn như bỏ nhầm, ngoài ra điều này giúp hạn chế lỗi khi làm nhiệm vụ.

Dễ dàng kiểm soát tiến độ nhiệm vụ, bởi vì hệ thống không cần kiểm tra túi đồ người chơi một cách phức tạp. Các vật phẩm nhiệm vụ sẽ nằm trong một luồng xử lý riêng, giúp theo dõi trạng thái nhiệm vụ một cách chính xác hơn.

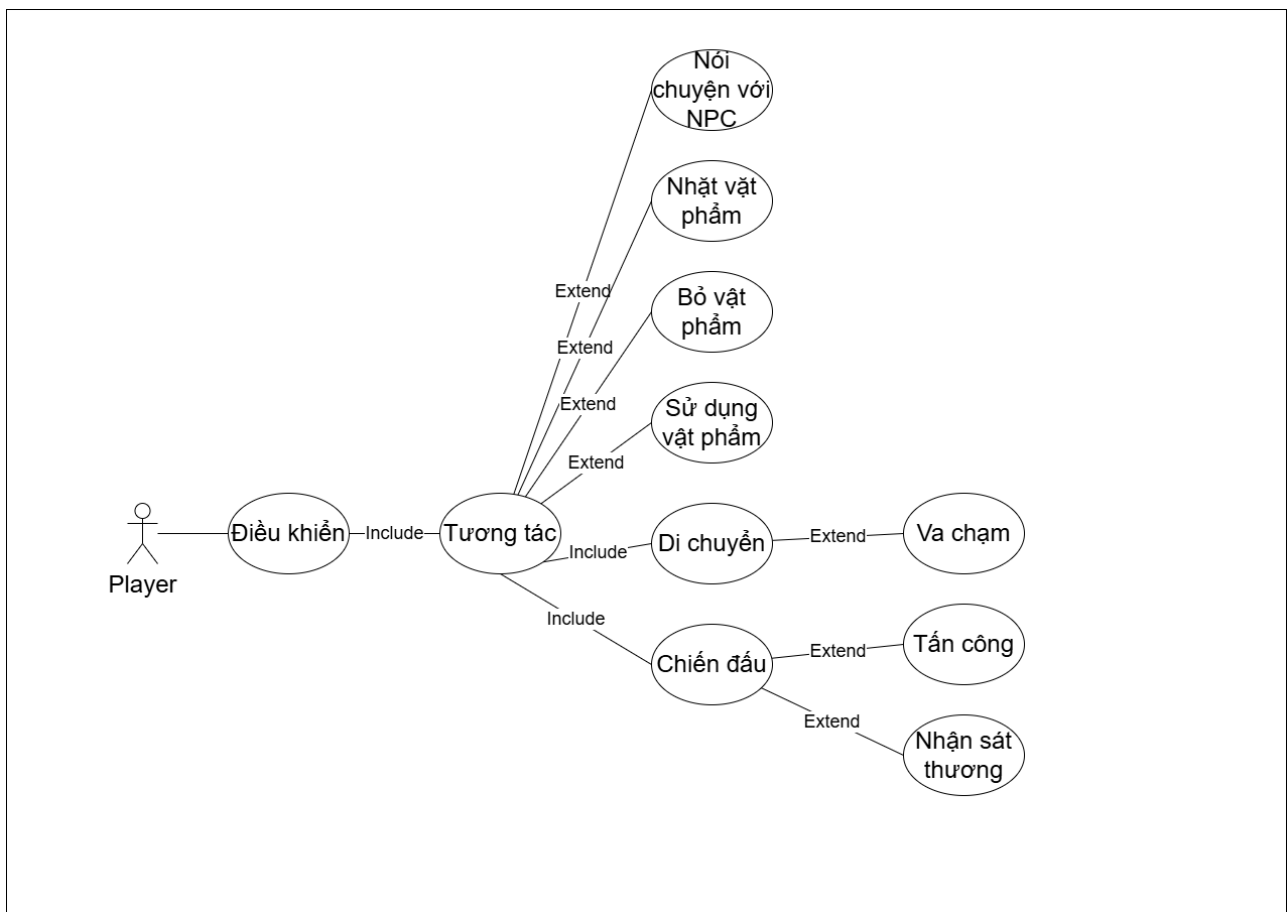


**Bảng 3.4 Cấu trúc dữ liệu lưu trữ dữ liệu vật phẩm**

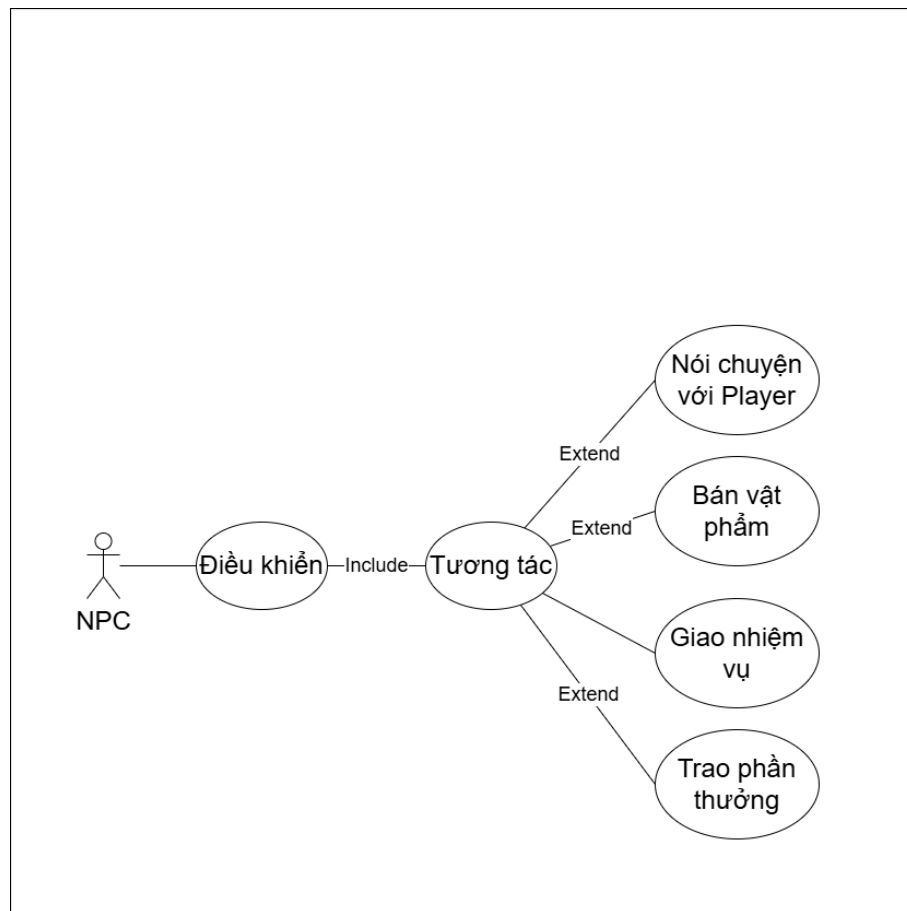
Thuộc tính	Kiểu dữ liệu	Ý nghĩa / Vai trò
item_id	String	Mã định danh duy nhất của vật phẩm. Hệ thống dùng ID để phân biệt vật phẩm và liên kết với dữ liệu mô tả, icon, thông số,...
item_name	String	Tên vật phẩm hiển thị trong UI túi đồ, cửa hàng hoặc khi nhặt item.
description	String	Mô tả chi tiết về vật phẩm (tác dụng, nguồn gốc, hiệu ứng...). Giúp người chơi hiểu công dụng khi hover hoặc mở menu xem chi tiết.
item_type	String	Loại vật phẩm, ví dụ: "consumable", "equipment", "material", "quest_item". Dùng để hệ thống xử lý logic khác nhau theo từng loại.
icon_path	String	Đường dẫn tới sprite/icon vật phẩm (asset 16x16 tự tạo). Giúp hệ thống hiển thị đúng hình vật phẩm trong UI inventory hoặc shop.
stackable	bool	Vật phẩm có thể chồng số lượng hay không. Ví dụ: potion thì được, vũ khí hoặc áo giáp thì không.
max_stack	int	Số lượng tối đa một ô chứa có thể chứa. Chỉ sử dụng nếu stackable = true.
value	int	Giá trị bán/mua vật phẩm trong hệ thống shop. Cho phép tương tác mua bán cơ bản.

effects	Dictionary	Danh sách hiệu ứng mà vật phẩm tạo ra khi sử dụng, ví dụ: { "hp_restore": 20 }. Phù hợp cho consumable.
is_equipable	bool	Xác định vật phẩm có thể trang bị hay không (vũ khí, giáp,...).
equip_slot	String	Loại slot mà item dùng khi trang bị: "weapon", "armor", "ring",... Chỉ dùng khi is_equipable = true.

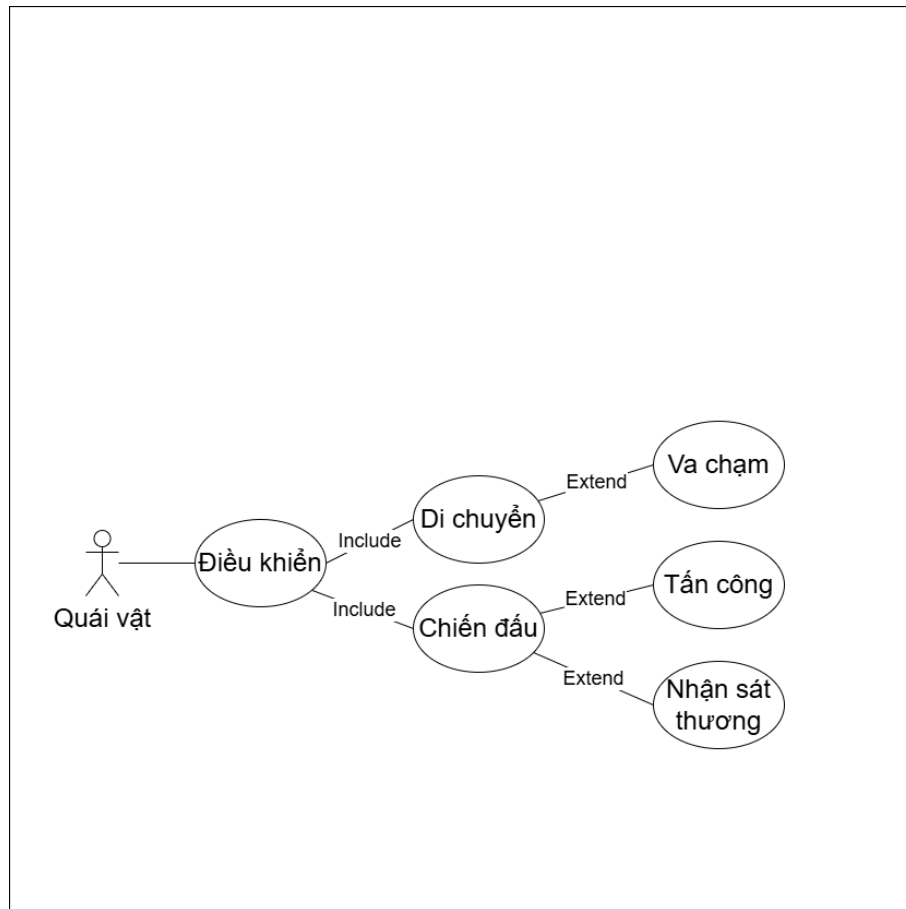
### 3.2.3. Thiết kế xử lý



**Hình 3.1 Sơ đồ Use Case của Actor Player**

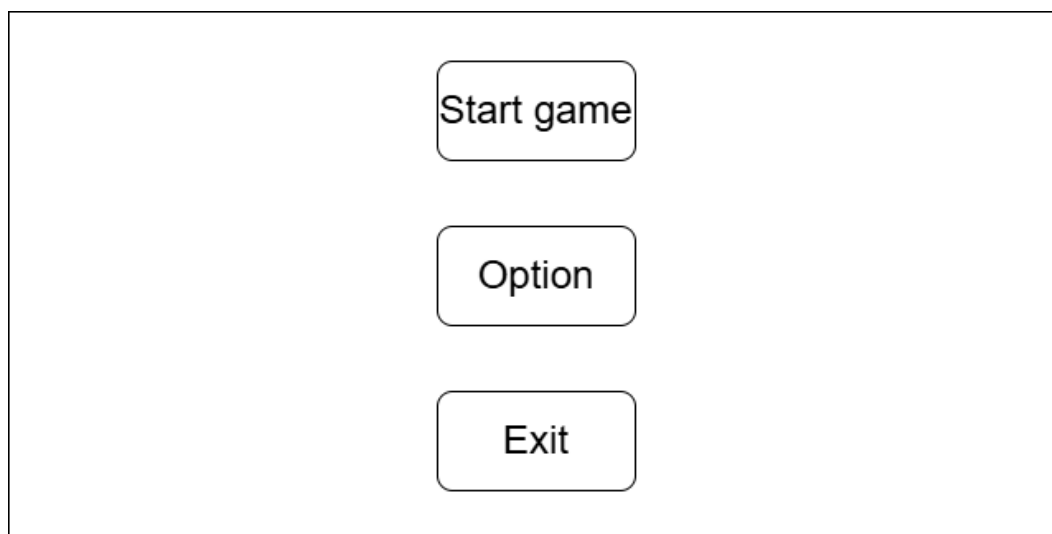


**Hình 3.2 Sơ đồ User Case của Actor NPC**

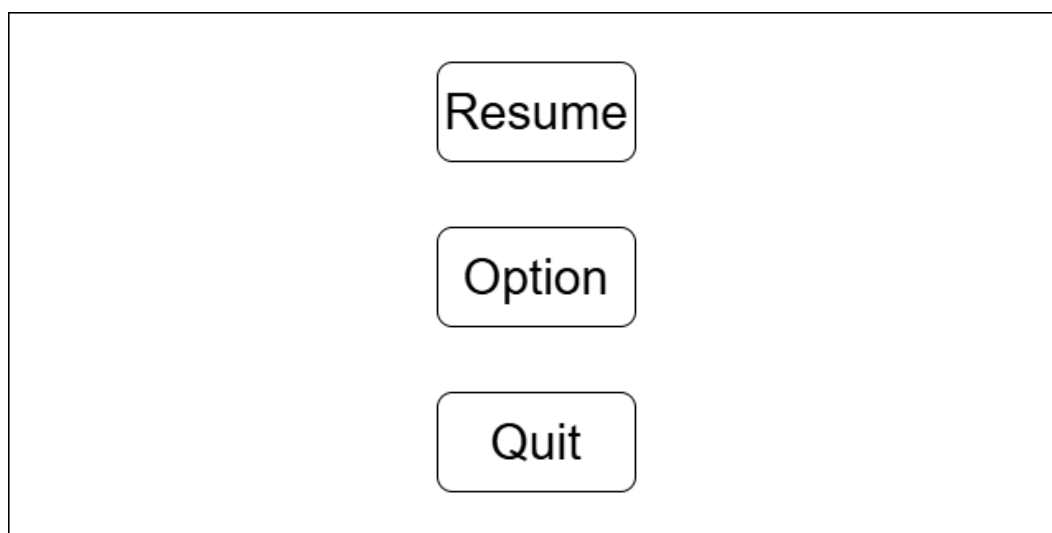


***Hình 3.3 Sơ đồ Use Case của Actor Quái vật***

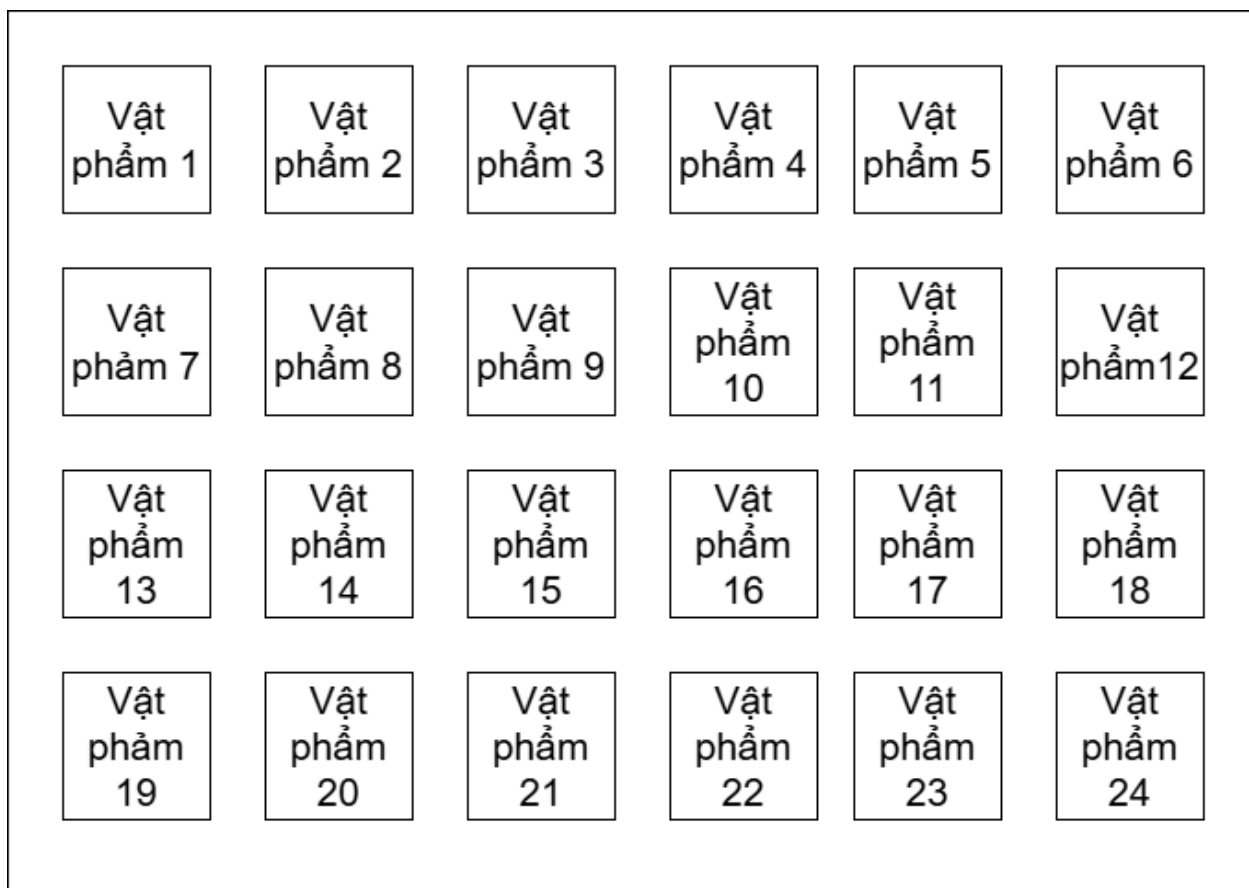
### 3.2.4. Thiết kế giao diện



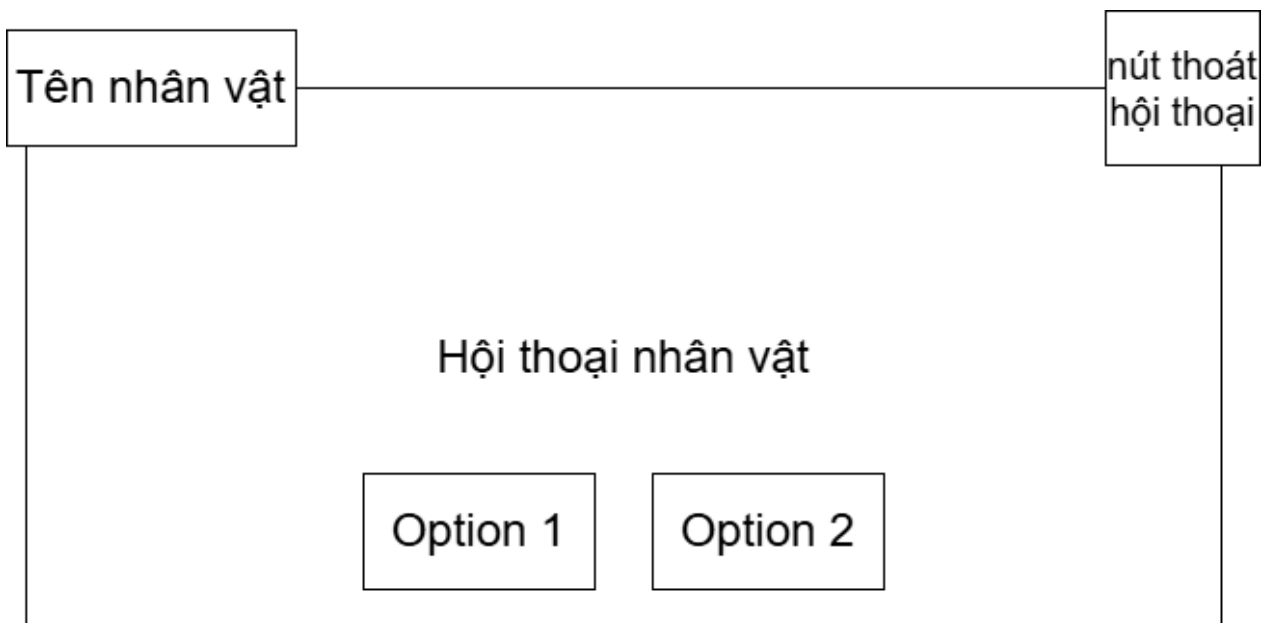
*Hình 3.4 Phác thảo giao diện màn hình chính*



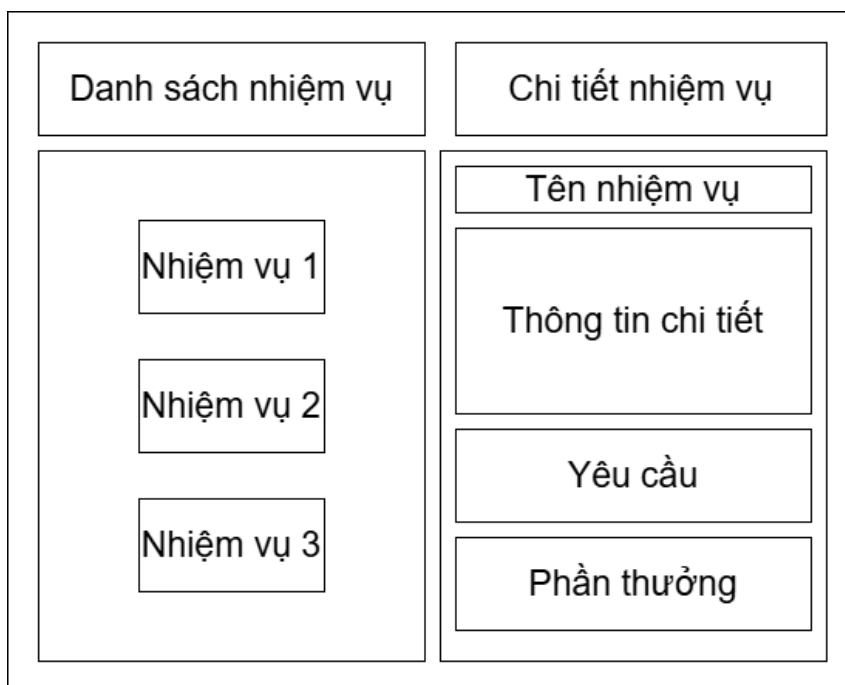
*Hình 3.5 Phác thảo giao diện màn hình tạm dừng game*



*Hình 3.6 Phát thảo giao diện túi đồ*



**Hình 3.7** *Phát thảo giao diện hội thoại*

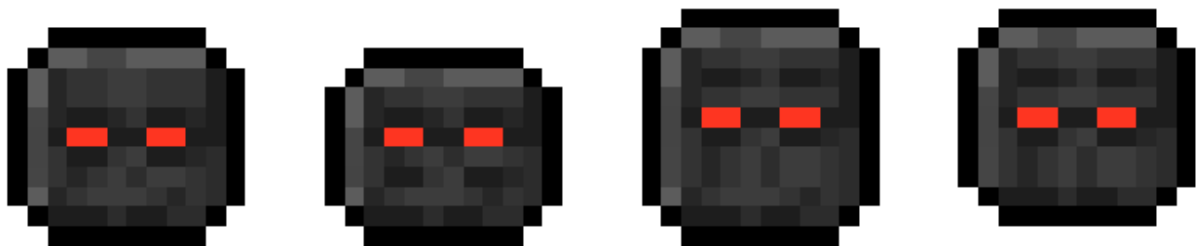


**Hình 3.8** *Phát thảo giao diện danh sách nhiệm vụ*

### 3.2.5. Thiết kế asset

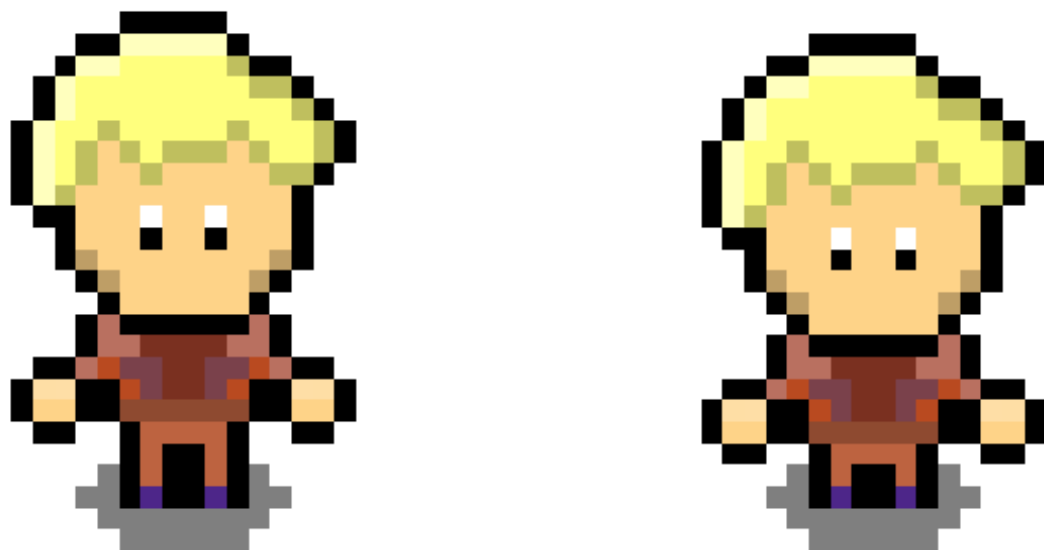


*Hình 3.9 Hoạt ảnh chuyển động của Player*

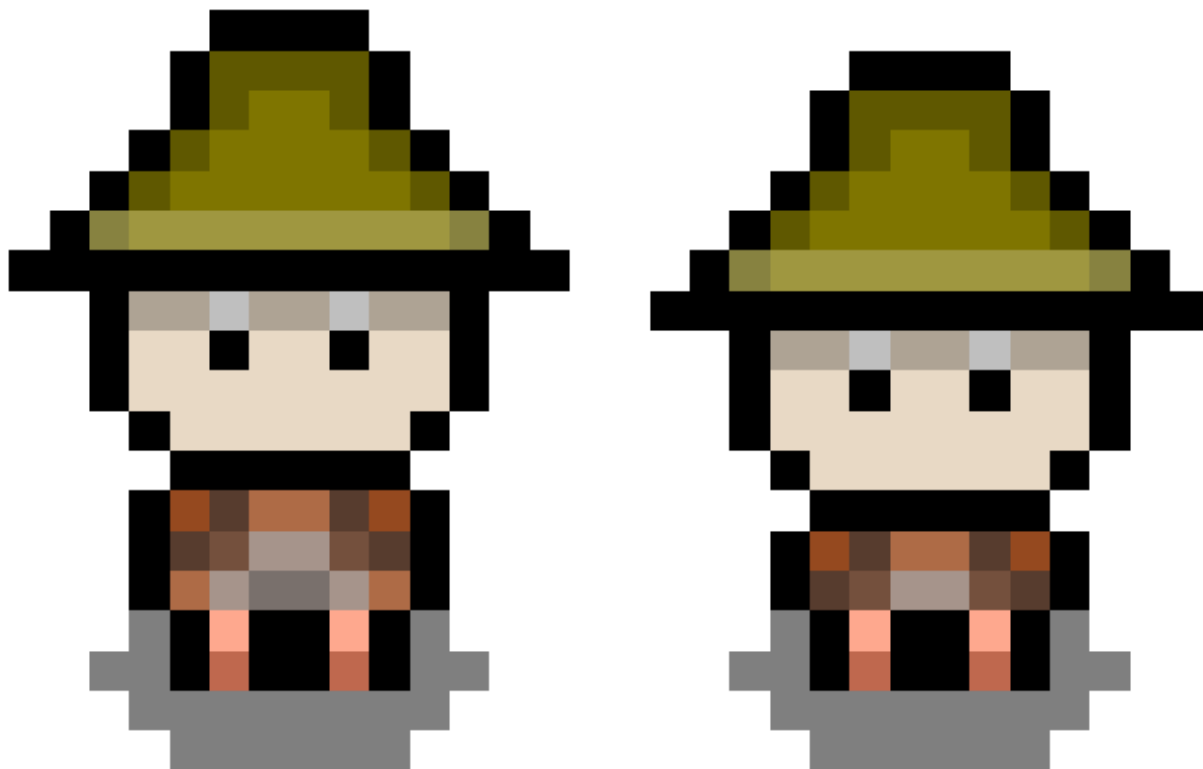


*Hình 3.10 Hoạt ảnh chuyển động của kẻ địch*





*Hình 3.11 Hoạt ảnh chuyển động của NPC hội trưởng*



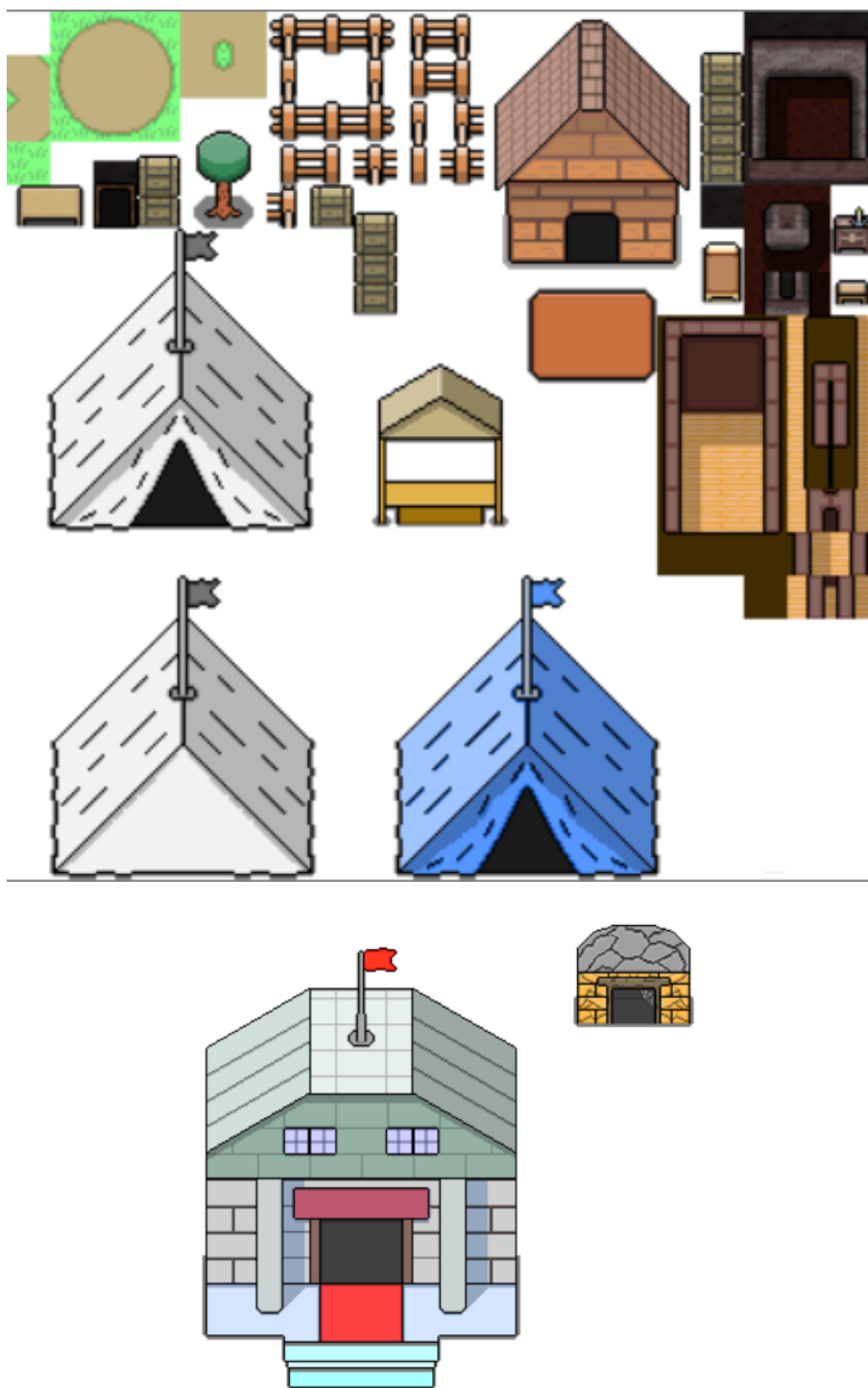
*Hình 3.12 Hoạt ảnh chuyển động của trưởng làng*



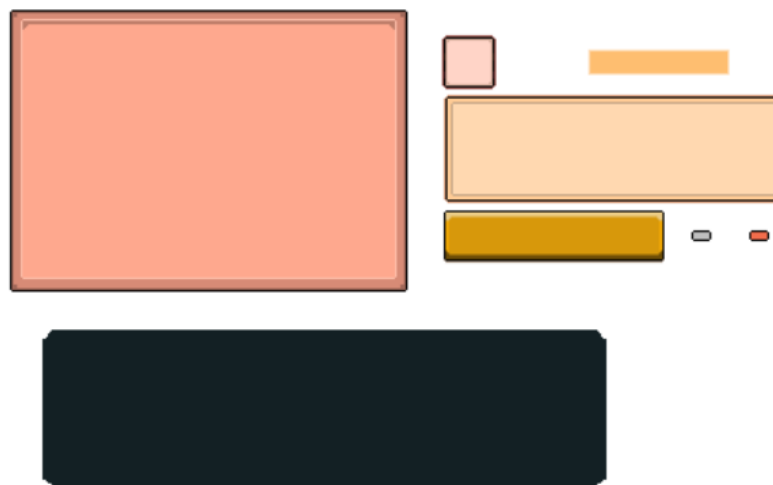
*Hình 3.13 Hoạt ảnh chuyển động của người bán hàng*



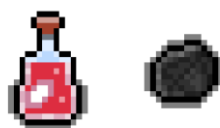
*Hình 3.14 Hoạt ảnh chuyển động của lính canh*



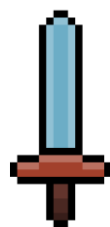
*Hình 3.15 Tileset xây dựng môi trường*



*Hình 3.16 Các khung để xây dựng giao diện hội thoại và túi đồ*



*Hình 3.17 Vật phẩm*



*Hình 3.18 Kiếm*



*Hình 3.19 Hiệu ứng nổ*

## CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU

### 4.1. Dữ liệu thử nghiệm

Quá trình đánh giá được thực hiện dựa trên các tệp dữ liệu mẫu. Các dữ liệu thử nghiệm được xây dựng theo đúng cấu trúc mà hệ thống sử dụng, bao gồm dữ liệu đối thoại (JSON), dữ liệu nhiệm vụ (Resource) và dữ liệu vật phẩm. Những dữ liệu này được sử dụng nhằm kiểm tra độ ổn định của các hệ thống chính trong game: di chuyển, va chạm, quản lý vật phẩm, tương tác NPC và nhiệm vụ.

#### 4.1.1. Dữ liệu đối thoại thử nghiệm (NPC Dialog JSON)

Bộ dữ liệu gồm 3 NPC: Trưởng làng, Hội trưởng, Người bán hàng. Mỗi NPC có một hoặc nhiều “tree” hội thoại và nhiều trạng thái (start, exit, buy, give\_quest). Mục đích sử dụng dữ liệu này trong thử nghiệm:

- Kiểm tra hệ thống đọc JSON và tạo hội thoại động.
- Kiểm tra tính hợp lệ của hướng chuyển trạng thái.
- Kiểm tra đường dẫn lựa chọn (branch) khi người chơi nhấn nút tương tác.

***Bảng 4.1 Bảng dữ liệu dialog***

NPC	Branch ID	State	Text	Options
Trưởng làng	chào hỏi	start	Xin chào.	Rời đi → exit
Trưởng làng	chào hỏi	exit		
Trưởng làng	chào hỏi	end		
Trưởng làng	npc_default	start	Chào cậu.	Rời đi → exit

Hội trưởng	nhận nhiệm vụ	start	Xin chào.	Nhận nhiệm vụ → give_quest Rời đi → exit
Hội trưởng	nhận nhiệm vụ	give_quest	Kiểm cho tôi 3 cục than.	Rời đi → exit
Người bán hàng	chào hỏi	start	Cậu muốn mua gì?	Rời đi → exit Thuốc hồi máu (50 gold) → buy_health_potion
Người bán hàng	chào hỏi	buy_health_potion	Cậu có muốn mua bình hồi máu với giá 50 gold không?	Mua → buy_health_potion Không mua → start Rời đi → exit

#### 4.1.2. Dữ liệu thử nghiệm nhiệm vụ (Quest Resource)

Các file Resource được tạo với danh sách Objective và Reward:

- Objective thuộc loại Collection tức là người chơi sẽ phải nhặt 3 cục than.
- Reward bao sẽ là Coin mà player nhận được sau khi hoàn thành nhiệm vụ.

Dữ liệu dùng để:

- Kiểm tra cập nhật tiến trình nhiệm vụ theo từng Objective.
- Kiểm tra trạng thái nhiệm vụ khi hoàn thành tất cả mục tiêu.
- Kiểm tra phản hồi của UI nhiệm vụ khi thay đổi trạng thái.



**Bảng 4.2 Bảng dữ liệu nhiệm vụ**

Thuộc tính	Giá trị
Quest ID	nhiem_vu_1
Quest Name	Thu thập
Description	Kiểm cho thị trường 3 cục than, sau khi hoàn thành bạn sẽ nhận 3 coin
State	in_progress
Repeatable	true
Unlock ID	nhận nhiệm vụ

**Bảng 4.3 Bảng dữ liệu mục tiêu của nhiệm vụ**

Objective ID	Description	Target Type	Required / Collected	Completed
tim_than	Đi thu thập than	Collection	0 / 3	false

**Bảng 4.4 Bảng dữ liệu phần thưởng của nhiệm vụ**

Reward Type	Reward Amount
coins	3

#### 4.1.3. Dữ liệu vật phẩm thử nghiệm (Inventory Item)

Dữ liệu mẫu gồm:

- Vật phẩm tiêu thụ: Bình máu
- Vật phẩm nhiệm vụ: Than đá

Kịch bản thử nghiệm:

- Thử nhặt vật phẩm thường để kiểm tra vật phẩm có thêm vào túi đồ không.
- Thử nhặt vật phẩm thuộc nhiệm vụ để xem hệ thống quản lý nhiệm vụ có cập nhật tiến trình nhiệm vụ thay vì thêm vào túi đồ hay không.
- Thử mua vật phẩm từ NPC bán hàng và kiểm tra xem vật phẩm đã mua có trong túi đồ hay chưa và tiền có trừ hay không.
- Thử kiểm tra tình huống túi đồ đầy.

***Bảng 4.5 Bảng dữ liệu vật phẩm***

Item Type	Item Name	Item Texture	Item Effect
Vật liệu	Than đá	atlas_texture	(trống)
Tiêu hao	Bình máu	atlas_texture	heal

## **4.2. Kết quả thực nghiệm**

### **4.2.1. Kết quả kiểm thử di chuyển, va chạm và camera**

Nhân vật di chuyển mượt ở 8 hướng, không xảy ra hiện tượng trượt (sliding) tại các vị trí hẹp. Va chạm với tường, vật cản và quái vật hoạt động đúng theo Collider trong Tileset. Camera2D theo dõi nhân vật ổn định, không rung và không lệch vị trí.

### **4.2.2. Kết quả kiểm thử hệ thống đối thoại NPC**

Toàn bộ dữ liệu JSON được load đúng. Khi người chơi chọn một option, hệ thống chuyển sang đúng state được định nghĩa trong JSON. Đối thoại dừng đúng khi chọn state “Rời đi”, không phát sinh lỗi.

#### **4.2.3. Kết quả kiểm thử hệ thống nhiệm vụ (Quest)**

Khi nhặt đúng vật phẩm, hệ thống cập nhật tiến trình nhiệm vụ đúng và đánh dấu hoàn thành nếu đủ số lượng. Khi hoàn thành toàn bộ mục tiêu của nhiệm vụ thì nhiệm vụ chuyển sang trạng thái hoàn thành. UI nhiệm vụ hiển thị đầy đủ tiến trình theo thời gian thực và không gặp lỗi.

#### **4.2.4. Kết quả kiểm thử hệ thống vật phẩm và túi đồ (Inventory)**

Khi nhặt vật phẩm thường thì vật phẩm sẽ thêm vào túi đồ đúng số lượng còn khi nhặt vật phẩm nhiệm vụ thì vật phẩm không thêm vào túi đồ mà cập nhật trực tiếp vào tiến trình của nhiệm vụ. Mua vật phẩm từ người bán hàng thì tiền trừ đúng số lượng khi mua. Hiển thị thông báo khi không đủ tiền.

#### **4.2.5. Kết quả kiểm thử chiến đấu cơ bản**

Người chơi có thể tấn công gần, gây sát thương chuẩn cho quái.

Quái vật nhận sát thương và hiển thị animation trúng đòn đúng thời điểm.

Quái truy đuổi người chơi trong phạm vi phát hiện.

Khi người chơi bị quái đụng trúng thì sẽ hiển thị animation trúng đòn và mất máu.

#### **4.2.6. Kết quả kiểm thử âm thanh và nhạc nền**

Nhạc nền load theo từng khu vực.

Âm thanh tấn công, nhặt đồ, hoạt động đúng thời điểm.

Không xuất hiện tình trạng trùng âm hoặc bị mất âm thanh.

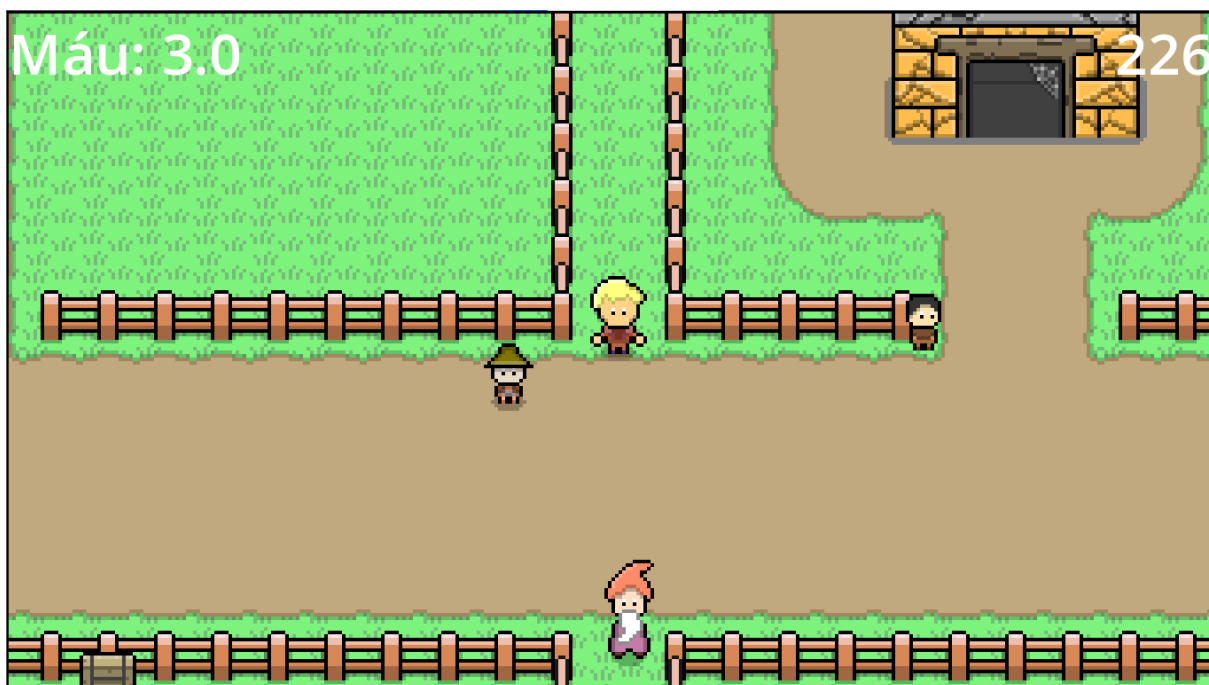
#### **4.2.7. Kết quả kiểm thử lưu và tải tiến trình**

Dữ liệu lưu ra JSON đầy đủ. Khi tải lại game, trạng thái, vị trí, map hiện tại, trạng thái nhiệm vụ và vật phẩm trong túi đồ của nhân vật được khôi phục hoàn toàn, không xảy ra trường hợp dữ liệu bị lỗi hoặc mất mát.

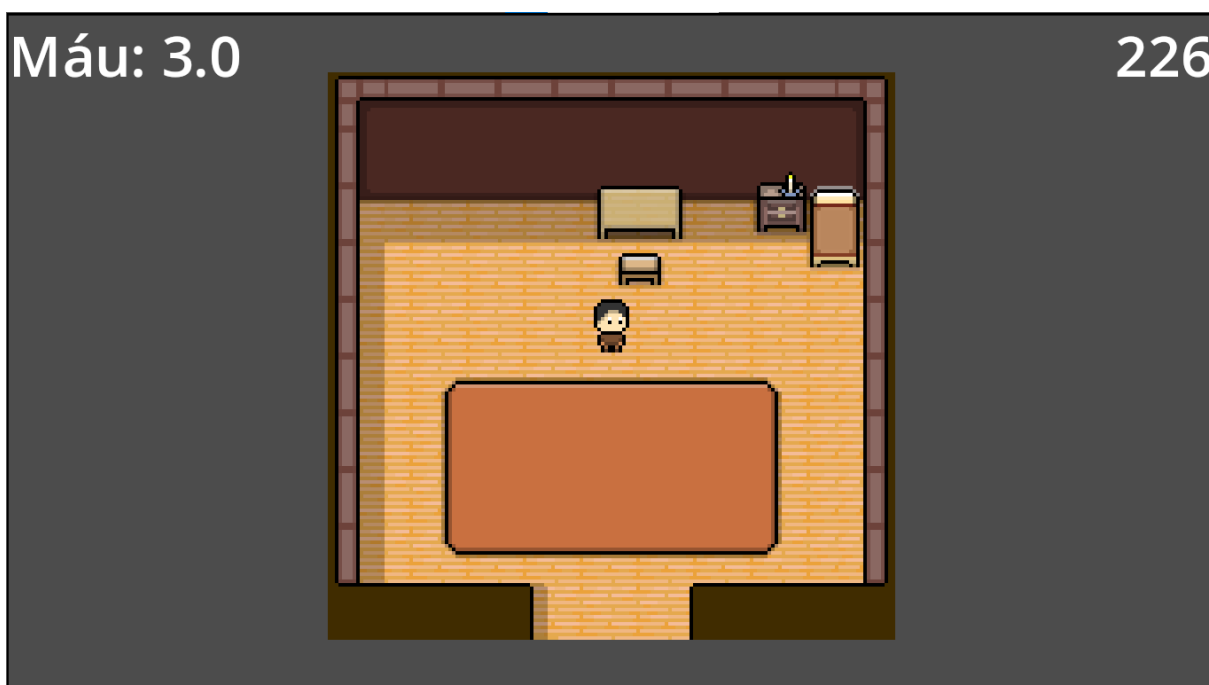
### 4.3. Giao diện và gameplay



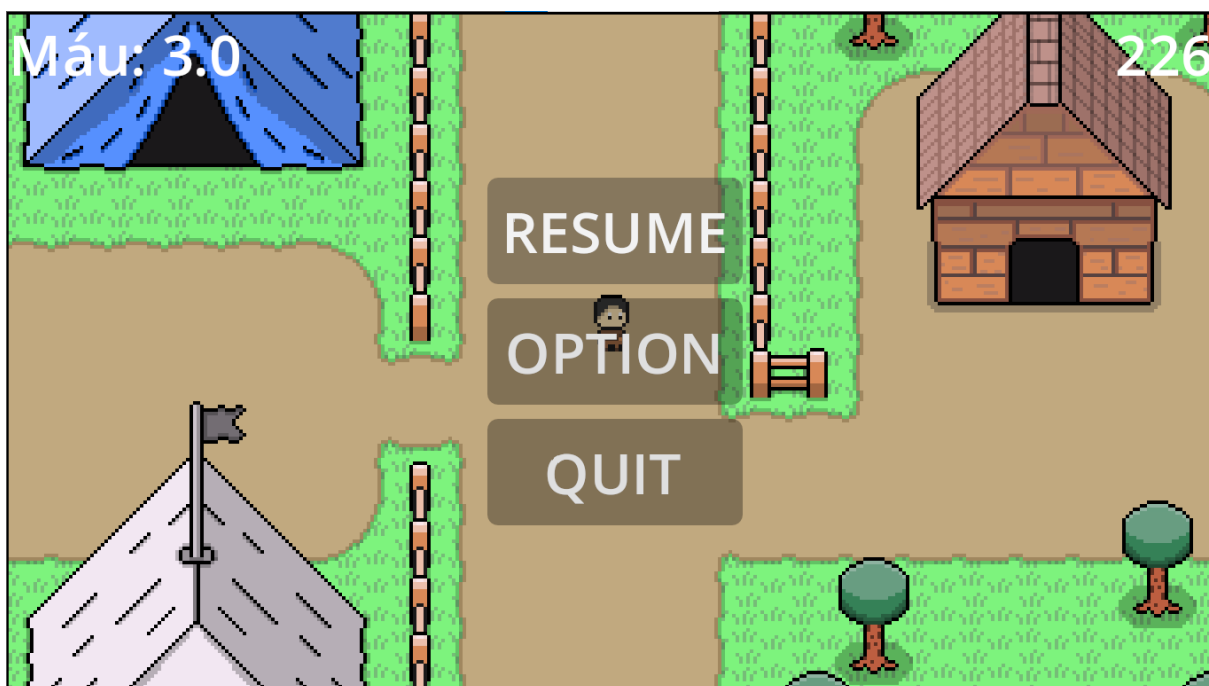
*Hình 4.1 Giao diện trang chủ*



*Hình 4.2 Giao diện gameplay của game*



*Hình 4.3 Player di chuyển sang map khác*



*Hình 4.4 Giao diện tạm dừng game*



Hình 4.5 Giao diện hộp thoại giao tiếp giữa player và NPC



Hình 4.6 Đoạn hội thoại để player nhận nhiệm vụ



Hình 4.7 Player nhận nhiệm vụ thành công

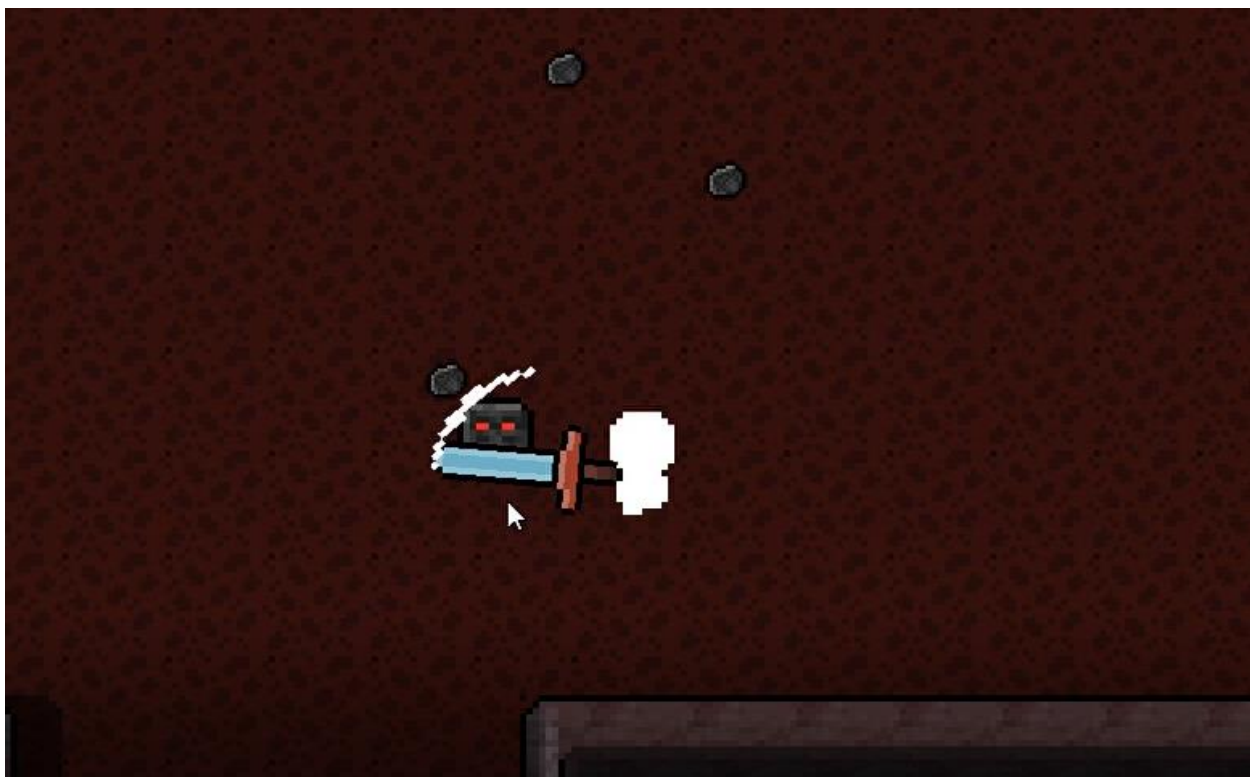


Hình 4.8 Giao diện danh sách nhiệm vụ

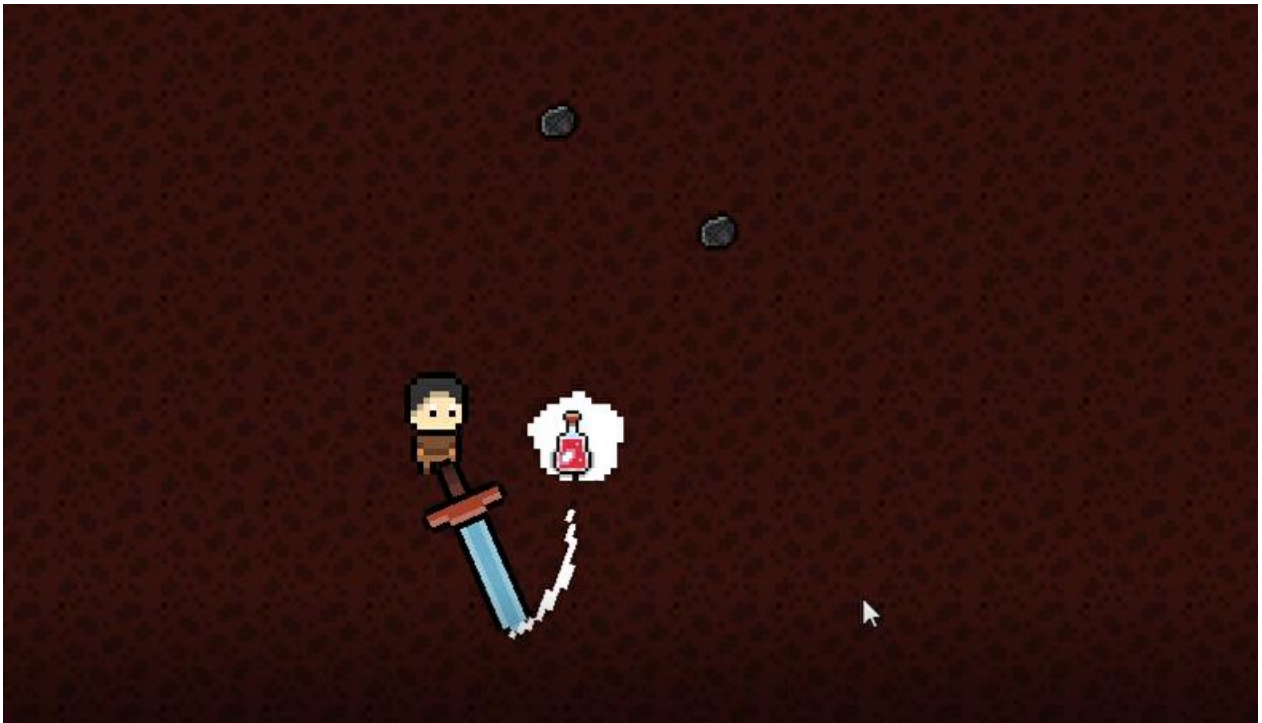




*Hình 4.9 Player tấn công kẻ địch*



*Hình 4.10 Player nhận sát thương từ kẻ địch khi bị đánh trúng*



*Hình 4.11 Kẻ địch rơi ra vật phẩm khi bị hạ gục*



*Hình 4.12 Khi người chơi đến gần vật phẩm, giao diện sẽ chỉ dẫn cách nhặt vật phẩm*



*Hình 4.13 Khi người chơi nhặt đúng vật phẩm cần thiết cho nhiệm vụ thì tiến trình trong giao diện nhiệm vụ sẽ tăng lên*



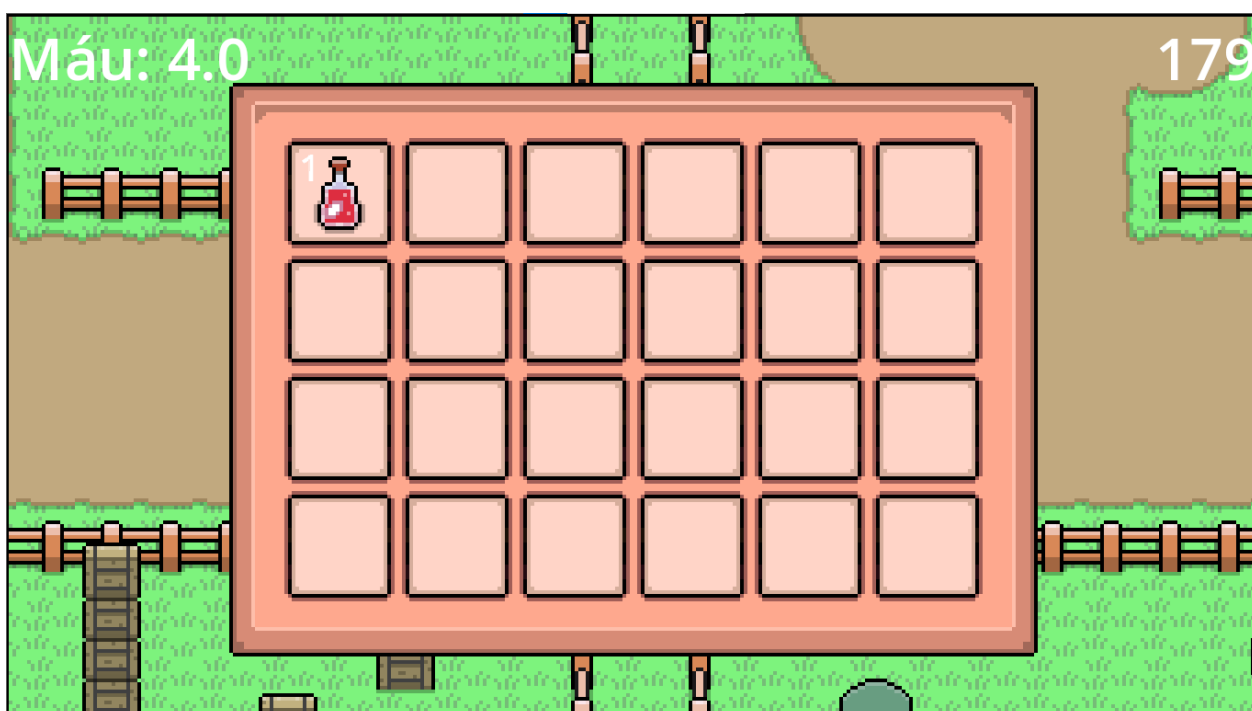
*Hình 4.14 Khi hoàn thành nhiệm vụ thì người chơi sẽ nhận được 3 coin cộng vào ở góc trên bên phải màn hình*



Hình 4.15 Player giao tiếp với người bán hàng



Hình 4.16 Player mua vật phẩm thành công



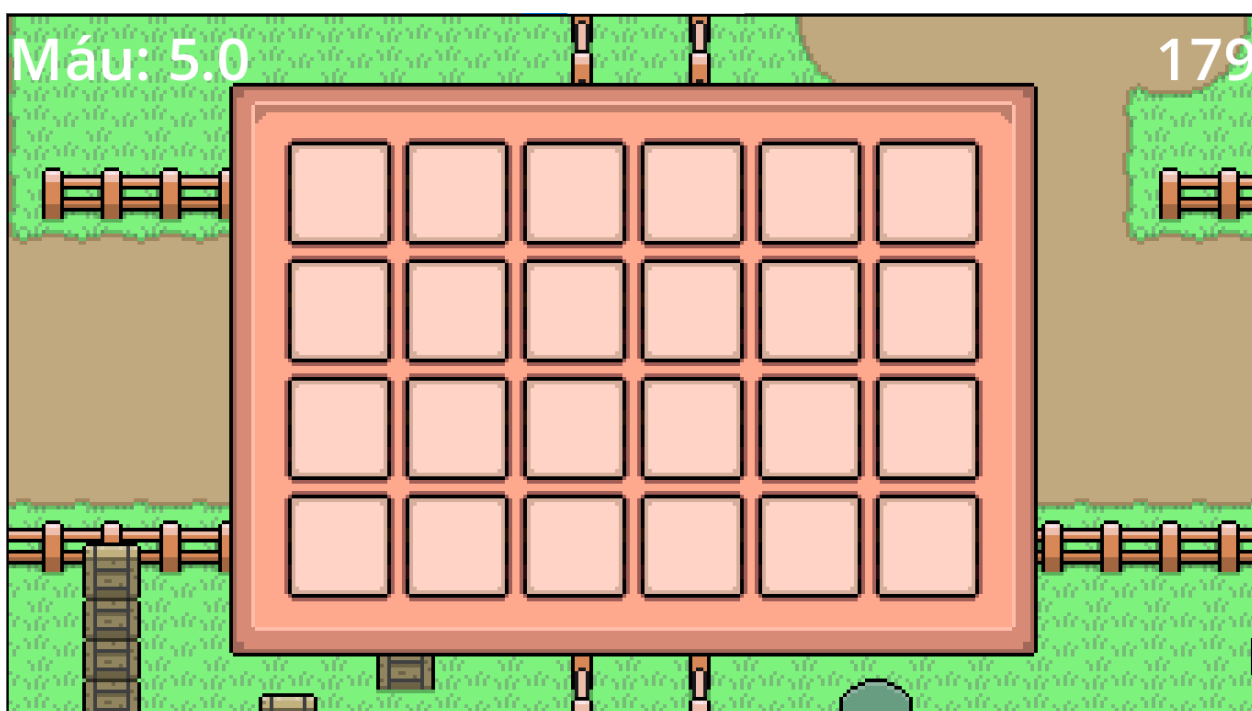
Hình 4.17 Giao diện túi đồ của Player



Hình 4.18 Giao diện hiển thị thông tin chi tiết về vật phẩm



*Hình 4.19 Giao diện lựa chọn sử dụng hoặc bỏ vật phẩm*



*Hình 4.20 Player hồi máu thành công từ 4.0 sang 5.0*



## CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1. Kết luận

Đề tài “Thiết kế và phát triển game nhập vai 2D pixel top-down bằng Godot Engine” đã được hoàn thành một cách đầy đủ và chính xác theo đúng mục tiêu ban đầu mà cá nhân đã đặt ra trước khi bắt đầu triển khai. Mục tiêu này bao gồm việc xây dựng một hệ thống gameplay cơ bản, đảm bảo rằng các chức năng nền tảng của một trò chơi nhập vai 2D được hiện thực hóa, đồng thời tạo ra một cấu trúc dữ liệu rõ ràng, dễ quản lý, dễ theo dõi và có khả năng mở rộng trong tương lai nếu muốn thêm các tính năng mới, nâng cấp đồ họa hoặc phát triển các phiên bản tiếp theo của trò chơi.

Thông qua quá trình thực hiện dự án đã áp dụng và thực hành những kiến thức đã học về lập trình game, thiết kế đồ họa pixel, tổ chức dữ liệu, quản lý scene, quản lý node, cũng như các kỹ thuật xử lý sự kiện trong môi trường Godot Engine. Quá trình này không chỉ giúp củng cố lý thuyết mà còn nâng cao khả năng giải quyết vấn đề thực tế, quản lý dự án và phối hợp giữa các thành phần trong game một cách hiệu quả.

Sản phẩm cuối cùng của đề tài cho phép người chơi điều khiển nhân vật chính một cách linh hoạt trong các môi trường khác nhau, bao gồm cả làng và hầm mỏ. Nhân vật có thể di chuyển mượt mà theo bốn hướng cơ bản, đồng thời hệ thống va chạm được triển khai chính xác, giúp nhân vật không đi xuyên qua các vật cản, tường hoặc các đối tượng khác trong game. Camera được thiết lập để theo dõi nhân vật liên tục, đảm bảo người chơi luôn quan sát được vị trí và hành động của nhân vật trong môi trường.

Hệ thống bản đồ và TileMap được thiết kế theo tái sử dụng, nghĩa là mỗi phân bản đồ, mỗi tile và mỗi khu vực có thể được tạo ra, chỉnh sửa và tái sử dụng một cách linh hoạt. Cách tổ chức này không chỉ giúp quá trình xây dựng môi trường game trở nên dễ dàng, nhanh chóng mà còn tạo điều kiện thuận lợi cho việc mở rộng hoặc thay đổi bản đồ trong tương lai mà không cần viết lại toàn bộ code hoặc xây dựng lại các tile từ đầu.

Hệ thống NPC, hội thoại và nhiệm vụ được thiết kế dựa trên các tệp JSON và Resource của Godot. Việc này giúp nội dung nhiệm vụ, hội thoại và các đặc tính của NPC trở nên dễ chỉnh sửa, dễ quản lý và dễ mở rộng về sau. Người phát triển có thể thêm, bớt hoặc sửa đổi nhiệm vụ và hội thoại mà không cần thay đổi cấu trúc code chính, giúp tiết kiệm thời gian và tăng tính linh hoạt.

Các hệ thống khác của game cũng đã được triển khai đầy đủ và hoạt động ổn định, bao gồm: túi đồ (inventory) để quản lý vật phẩm với nhiều loại vật phẩm khác nhau, cơ chế mua bán giữa người chơi và cửa hàng NPC, thu thập tài nguyên, chiến đấu cơ bản, âm thanh và nhạc nền, cũng như giao diện người dùng (UI). Mỗi hệ thống được thiết kế sao cho hoạt động đồng bộ với các thành phần khác, đảm bảo trải nghiệm chơi game liền mạch, trực quan và dễ sử dụng.

Ngoài ra, cơ chế lưu và tải dữ liệu bằng JSON cũng đã được triển khai thành công. Người chơi có thể thoát game bất kỳ lúc nào mà không sợ bị mất dữ liệu tiến trình đang chơi, điều này đặc biệt quan trọng trong các trò chơi nhập vai, nơi tiến trình và vật phẩm của người chơi cần được bảo toàn giữa các phiên chơi khác nhau. Việc sử dụng JSON giúp dữ liệu lưu trữ có cấu trúc rõ ràng, dễ đọc, dễ chỉnh sửa và dễ mở rộng trong các phiên bản tiếp theo.

Về tổng thể, đề tài đã thành công trong việc xây dựng một nền tảng RPG 2D hoàn chỉnh ở mức cơ bản nhưng vẫn đầy đủ các chức năng cần thiết. Nền tảng này tạo ra một cấu trúc game vững chắc, có thể sử dụng làm cơ sở để phát triển các phiên bản nâng cao hơn trong tương lai, như thêm nhiều loại quái vật, nhiệm vụ phức tạp hơn, hệ thống kỹ năng hoặc cải thiện đồ họa. Nền tảng này cũng chứng minh khả năng áp dụng Godot Engine để thiết kế và phát triển một trò chơi nhập vai 2D từ cơ bản đến nâng cao, đồng thời cung cấp trải nghiệm chơi game thú vị và ổn định cho người chơi.

## 5.2. Hạn chế

Mặc dù đề tài đã hoàn thành các chức năng cốt lõi và đạt được các mục tiêu ban đầu về xây dựng một game nhập vai 2D pixel top-down với các hệ thống gameplay cơ bản, vẫn tồn tại một số hạn chế và điểm cần cải thiện, nhưng đồng thời các hạn chế này cũng là để rút kinh nghiệm cho bản thân, từ đó có thể áp dụng kiến thức đã học để cải thiện và phát triển các dự án tiếp theo trong phạm vi rộng hơn và quy mô lớn hơn, phức tạp hơn.

### 5.2.1. Gameplay còn đơn giản

Hiện tại, hệ thống chiến đấu trong game mới chỉ đạt mức cơ bản, chỉ cho phép nhân vật di chuyển và tấn công một cách đơn giản mà chưa có nhiều tính năng phức tạp. Chẳng hạn, game chưa có các loại vũ khí khác nhau với đặc tính và sát thương riêng biệt, chưa có hiệu ứng kỹ năng như phép thuật hay đòn tấn công đặc biệt; AI quái vật cũng còn đơn giản, chỉ thực hiện các hành vi đơn giản như đuổi theo người chơi mà chưa có logic né tránh, phản đòn, phối hợp tấn công hay thay đổi chiến thuật theo tình huống. Ngoài ra, game cũng chưa triển khai các cơ chế nâng cao như né tránh, phòng thủ, hay combo đòn đánh, khiến trải nghiệm chiến đấu chưa thực sự đa dạng và còn cảm giác đơn điệu.

### 5.2.2. Thiếu chiều sâu nội dung

Game hiện tại thiếu tuyến truyện chính, chưa có bối cảnh cốt truyện đầy đủ hay các sự kiện dẫn dắt nhân vật qua nhiều giai đoạn khác nhau. Hệ thống tiến cấp nhân vật cũng chưa được triển khai, do đó người chơi không thể nâng cấp kỹ năng, chỉ số hay khả năng chiến đấu của nhân vật theo thời gian. Ngoài ra, hệ thống nhiệm vụ chưa đa dạng: các nhiệm vụ hiện tại chủ yếu là thu thập vật phẩm hoặc trò chuyện với NPC, chưa có nhiệm vụ nhiều giai đoạn, nhiệm vụ phân nhánh hay các nhiệm vụ phức tạp đòi hỏi nhiều bước để hoàn thành. Điều này dẫn đến việc game còn thiếu chiều sâu, tính thử thách và cảm giác khám phá, khiến trải nghiệm tổng thể chưa thực sự phong phú.

### 5.2.3. Hạn chế về đồ họa và hoạt ảnh

Do cá nhân tự thiết kế asset  $16 \times 16$ , số lượng khung hoạt ảnh còn hạn chế, nên các chuyển động của nhân vật và quái vật chưa thực sự mượt mà. Bản đồ hiện tại cũng còn ít biến thể về địa hình, dẫn đến cảm giác môi trường chưa đa dạng. Ngoài ra, các hiệu ứng ánh sáng, shader hoặc các chi tiết hình ảnh nâng cao chưa được áp dụng, khiến môi trường và các khu vực trong game còn đơn điệu và chưa tạo được bầu không khí sống động như các trò chơi RPG chuyên nghiệp.

### 5.2.4. Âm thanh còn tối giản

Về mặt âm thanh, game mới chỉ trang bị nhạc nền và hiệu ứng âm thanh cơ bản, chưa có các hệ thống nâng cao như điều chỉnh âm lượng theo bối cảnh, âm thanh môi trường (ambient sound) hay các hiệu ứng động khi di chuyển, chiến đấu. Điều này làm giảm phần nào trải nghiệm nhập vai, vì âm thanh đóng vai trò quan trọng trong việc tạo ra không khí, cảm xúc và sự sống động cho môi trường game.

### 5.2.5. AI NPC và quái vật đơn giản

Các NPC trong game hiện tại chỉ có đối thoại tĩnh, không có hành vi di chuyển hay thay đổi theo thời gian hoặc theo điều kiện trò chơi. Quái vật chỉ có một dạng hành vi duy nhất là đuổi theo người chơi, chưa có logic né tránh, phối hợp với đồng loại, sử dụng kỹ năng đặc biệt hoặc thay đổi chiến thuật tùy theo tình huống. Điều này khiến hệ thống tương tác và thử thách trong game còn hạn chế, chưa mang lại cảm giác sinh động và phản ứng theo cách tự nhiên.

### 5.2.6. Hệ thống lưu trữ dữ liệu còn thủ công

Hiện tại, việc lưu trữ dữ liệu JSON vẫn còn thực hiện thủ công, chưa được tối ưu hóa cho các dự án lớn. Điều này tiềm ẩn rủi ro lỗi dữ liệu nếu số lượng NPC, vật phẩm, nhiệm vụ hay bản đồ tăng lên hoặc khi muốn mở rộng nội dung game trong tương lai. Hệ thống lưu trữ hiện tại hoạt động ổn định ở mức cơ bản, nhưng nếu mở rộng quy mô, cần thiết kế

lại cơ chế lưu và tải dữ liệu để giảm thiểu rủi ro mất dữ liệu và tăng khả năng quản lý, đặc biệt khi game có nhiều thực thể hoặc tiến trình phức tạp được thêm vào.

### **5.3. Hướng phát triển**

Để có thể phát triển thiết kế hệ thống game thành một sản phẩm hoàn chỉnh hơn về mặt gameplay và mặt hình ảnh, dự án có thể được mở rộng theo những hướng sau. Đây đều là những hướng phát triển tự nhiên, hợp lý và tiếp nối trực tiếp từ những gì đề tài đã xây dựng trong giai đoạn hiện tại.

#### **5.3.1. Mở rộng gameplay và hệ thống chiến đấu**

Hệ thống chiến đấu là trung tâm của rất nhiều trò chơi nhập vai 2D, vì vậy có thể tiếp tục được mở rộng theo nhiều hướng khác nhau để tăng chiều sâu gameplay. Một trong những hướng phát triển rõ ràng nhất là bổ sung nhiều loại vũ khí khác nhau, chẳng hạn như kiếm, cung, gậy phép hoặc các vũ khí đặc thù theo kiểu chiến đấu. Mỗi loại vũ khí có thể đi kèm bộ chỉ số riêng, tạo ra sự lựa chọn và chiến lược cho người chơi.

Bên cạnh đó, có thể xây dựng hệ thống kỹ năng (skill tree), cho phép nhân vật mở khóa các chiêu thức đặc biệt, gia tăng sức mạnh theo thời gian hoặc tùy biến theo lối chơi cá nhân. Một hệ thống kỹ năng hoàn chỉnh sẽ giúp người chơi có cảm giác phát triển nhân vật rõ ràng hơn.

Ngoài ra, AI của quái vật có thể được cải thiện để chúng có nhiều hành vi hơn, chẳng hạn như tấn công tầm xa, thay đổi hành vi theo từng giai đoạn trong trận đánh hoặc thậm chí bổ sung boss với cơ chế chiến đấu riêng biệt. Điều này sẽ khiến các trận chiến trở nên sinh động, hấp dẫn và mang tính thử thách hơn.

#### **5.3.2. Hoàn thiện hệ thống nhiệm vụ**

Hệ thống nhiệm vụ hiện tại có thể mở rộng bằng việc phát triển thêm nhiều loại nhiệm vụ khác nhau, nhiệm vụ có điều kiện kích hoạt hoặc nhiệm vụ hàng ngày giúp game có chiều sâu và tiếp thêm động lực cho người chơi gắn bó với game lâu dài hơn.

Ngoài ra, có thể mở rộng theo hướng nhiệm vụ phân nhánh, nơi người chơi đưa ra lựa chọn khác nhau và mỗi lựa chọn có thể dẫn đến kết quả khác nhau, ảnh hưởng đến môi trường hoặc NPC. Điều này sẽ tăng tính nhập vai và tạo cảm giác thế giới game phản hồi lại hành động của người chơi.

### 5.3.3. Mở rộng thế giới game

Thay vì chỉ có làng và hầm mỏ, thế giới game có thể được mở rộng thêm nhiều khu vực mới như rừng, sa mạc, núi tuyết, hang động hoặc vùng đất bị nguyên rủa. Mỗi khu vực có thể sở hữu tile set riêng, quái vật riêng và bối cảnh riêng.

Bên cạnh đó, có thể áp dụng hệ thống thời tiết (mưa, sương mù, gió nhẹ) hoặc chu kỳ ngày và đêm, giúp môi trường game trở nên sống động hơn. Một số trò chơi còn thay đổi hành vi quái hoặc NPC tùy theo thời điểm, điều này cũng có thể áp dụng để tăng tính đa dạng và hấp dẫn cho gameplay.

### 5.3.4. Cải thiện đồ họa và hiệu ứng

Mặc dù phong cách pixel 16×16 đã phù hợp với quy mô của đề tài, có thể nâng cao chất lượng hình ảnh bằng cách chuyển sang tileset 32×32 hoặc tăng số khung hoạt ảnh để tạo cảm giác mượt mà hơn.

Shader cũng là một hướng nâng cấp quan trọng. Có thể áp dụng các hiệu ứng như ánh sáng giả (2D lighting), bóng đổ, outline, hay các hiệu ứng khi nhân vật trúng đòn, nhặt đồ, mở rương hoặc khi chuyển map. Những hiệu ứng nhỏ này góp phần nâng cao cảm giác chuyên nghiệp và sống động của trò chơi.

### 5.3.5. Tối ưu UI và trải nghiệm người chơi

Một số tính năng UI phổ biến trong RPG có thể được thêm vào, chẳng hạn như minimap để định hướng hoặc hệ thống hiển thị nhiệm vụ đang làm trên HUD.

Bên cạnh đó, việc xây dựng hệ thống hướng dẫn (tutorial) giúp người chơi mới dễ làm quen hơn với gameplay.

Các giao diện như inventory, shop và dialog cũng có thể được tinh chỉnh UX để thân thiện hơn, chẳng hạn sắp xếp vật phẩm thông minh, so sánh trang bị hoặc thêm hiệu ứng và âm thanh phản hồi khi thao tác.

### **5.3.6. Nâng cấp hệ thống lưu trữ**

Hệ thống lưu JSON hiện tại có thể được nâng cấp bằng cách kết hợp Resource và ConfigFile hoặc chuyển sang sử dụng SQLite nếu game phát triển thành dự án lớn hơn. Những công cụ này giúp quản lý dữ liệu phức tạp hiệu quả hơn.

Ngoài ra, có thể bổ sung cơ chế kiểm tra tính hợp lệ của dữ liệu, đồng bộ dữ liệu khi tải hoặc hỗ trợ nhiều file save cùng lúc.

### **5.3.7. Xuất bản đa nền tảng**

Có thể mở rộng phạm vi tiếp cận của game, chẳng hạn như tối ưu để xuất bản trên Android hoặc Web (HTML5).

Ngoài ra, có thể bổ sung hệ thống điều khiển bằng cảm ứng hoặc gamepad để người chơi trên điện thoại hoặc thiết bị console mini có thể thao tác dễ dàng và thoải mái hơn.

## TÀI LIỆU THAM KHẢO

- [1] "Godot Engine 4.5 documentation in English," [Online]. Available: <https://docs.godotengine.org/en/stable/>.
- [2] S. Maglione, "How to create a Pixel Art Tileset - Complete Guide," [Online]. Available: <https://www.sandromaglione.com/articles/how-to-create-a-pixel-art-tileset-complete-guide>.
- [3] A. M. a. t. G. E. c. Juan Linietsky, "Godot Engine 4.4 documentation in English," [Online]. Available: <https://docs.godotengine.org/en/4.4/about/introduction.html>.