TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO MÔN:

GAME DEVELOPMENT

FINAL PROJECT

Giảng viên hướng dẫn: ThS. **VŨ ĐÌNH HỒNG**
Người thực hiện: **LÊ TUẤN THÔNG - 51900562**
Khóa: **23**
LÂM MINH THÔNG - 51900836
Khóa: **23**
TRỊNH QUANG HUY - 52000838
Khóa: **24**

Thành phố Hồ Chí Minh, 2024.

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO MÔN:

GAME DEVELOPMENT

FINAL PROJECT

Giảng viên hướng dẫn: ThS. **VŨ ĐÌNH HỒNG**
Người thực hiện: **LÊ TUẤN THÔNG - 51900562**
Khóa: **23**
LÂM MINH THÔNG - 51900836
Khóa: **23**
TRỊNH QUANG HUY - 52000838
Khóa: **24**

Thành phố Hồ Chí Minh, 2024.

LỜI CẢM ƠN

Chúng em xin được gửi lời cảm ơn đến giảng viên Vũ Đình Hồng. Trong quá trình học tập và tìm hiểu bộ môn Phát triển trò chơi, chúng em đã học được rất nhiều kiến thức cũng như nhiều điều thú vị từ những tiết học của thầy. Thầy đã giúp chúng em mở rộng thêm nhiều kiến thức mới cũng như chuyên sâu về môn này. Trong quá trình học tập chúng em vẫn còn yếu kém nhưng từ những kiến thức được học, chúng em đã có thêm cho mình những kiến thức quan trọng về kỹ năng chuyên ngành cũng như kiến thức để tìm tòi và giải quyết vấn đề.

Trong quá trình học tập không tránh khỏi những thiếu sót, chúng em mong sẽ nhận được góp ý đến từ thầy để bài báo cáo của chúng em được hoàn thiện hơn.

Ngoài ra em xin cảm ơn phía nhà trường đã tạo điều kiện cho chúng em có một môi trường học tập tốt để chúng em có được thêm những kiến thức này.

Kính chúc thầy sức khỏe, hạnh phúc và thành công trên con đường sự nghiệp giảng dạy.

BÀI BÁO CÁO ĐƯỢC HOÀN THÀNH
TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Em xin cam đoan đây là bài báo cáo sản phẩm của chỉ riêng chúng tôi. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa được công bố dưới bất kỳ hình thức nào. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đề tài còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc rõ ràng và cụ thể.

Nếu phát hiện có bất kỳ sự gian lận nào em xin hoàn toàn chịu trách nhiệm về nội dung của bài. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền trong quá trình thực hiện của em.

Trường ĐH Tôn Đức Thắng,
Ngày 30 tháng 04 năm 2024.
Sinh viên thực hiện,
(Ký và ghi rõ họ tên)

LÊ TUẤN THÔNG .

LÂM MINH THÔNG.

TRỊNH QUANG HUY.

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ

Phần đánh giá của giảng viên chấm bài:

.....

.....

.....

.....

.....

.....

TP. Hồ Chí Minh, ngày..... tháng..... năm 2024.

Giảng viên chấm bài,

.....

Phần đánh giá của giảng viên hướng dẫn:

.....

.....

.....

.....

.....

.....

TP. Hồ Chí Minh, ngày..... tháng..... năm 2024.

Giảng viên hướng dẫn,

VŨ ĐÌNH HỒNG.

TÓM TẮT

1. Cốt truyện:

Trong thế giới ảo Rarel, vương quốc này đang chìm trong bóng tối do sự xâm chiếm của các thế lực tà ác. Nhà vua đã thất bại trong việc bảo vệ đất nước và công chúa của vương quốc đã bị bắt cóc. Trước nguy cơ mất mát và nguy hiểm trước mắt, một nông dân trẻ tuổi, mang trong mình ước mơ trở thành anh hùng, đã bước lên hành trình đánh bại thế lực tà ác và giải cứu vương quốc khỏi sự kiểm soát của chúng.

2. Cách chơi:

Người chơi sẽ đảm nhận vai trò của người anh hùng trẻ tuổi, điều khiển nhân vật di chuyển qua các cấp độ khác nhau trong vương quốc Rarel. Trên đường đi, người chơi sẽ phải đối mặt với các con quái vật và thủ lĩnh tà ác. Cuối mỗi cấp độ là một trận đấu với một Boss quái vật mạnh mẽ. Người chơi có thể thu thập đồng vàng để mua các vật phẩm, trang bị và nâng cấp kỹ năng.

3. Các nhân vật:

- Nhân vật chính: Người anh hùng trẻ tuổi, người sẽ dẫn dắt người chơi qua hành trình giải cứu vương quốc.

- Quái vật: Đây là những kẻ địch đáng sợ, có nhiều hình dạng và kỹ năng khác nhau, đặc biệt là Boss quái vật ở cuối mỗi cấp độ.

- Boss: Những con quái vật mạnh mẽ và khó khăn, yêu cầu sự chiến đấu thông minh và kỹ năng tốt từ người chơi.

- NPC: Các nhân vật không phải người chơi hoặc quái vật, nhưng có thể cung cấp hỗ trợ, nhiệm vụ hoặc thông tin quan trọng cho người chơi.

4. Vật phẩm:

- Đồng vàng: Được sử dụng để mua các vật phẩm hữu ích như potion hồi máu, trang bị và vật phẩm nâng cấp khác. Mỗi lượng đồng vàng thu thập được cũng có thể tăng lượng máu cho nhân vật của người chơi khi đạt một ngưỡng nhất định.

5. Hệ thống chiến đấu đa dạng:

- Người chơi có thể sử dụng một loạt các kỹ năng và trang bị để đối phó với các quái vật và Boss. Tính toán chiến thuật và sử dụng các phản ứng phù hợp là chìa khóa

để chiến thắng.

6. Cấp độ và thế giới rộng lớn:

- Game có nhiều cấp độ với môi trường đa dạng từ rừng sâu đến hang động bí ẩn. Mỗi cấp độ mang đến một trải nghiệm mới và thú vị.

7. Hệ thống nâng cấp:

- Người chơi có thể nâng cấp nhân vật của mình thông qua việc tăng cường kỹ năng, sức mạnh và sức bền, giúp họ trở nên mạnh mẽ hơn để đối phó với những thách thức khó khăn.

8. Trang bị và vật phẩm:

- Ngoài việc thu thập đồng vàng, người chơi cũng có thể tìm thấy và mua các loại trang bị và vật phẩm khác nhau trong game, từ vũ khí đến áo giáp, từ potion hồi máu đến item hỗ trợ.

9. Môi trường tương tác:

- Một số khu vực trong game có thể tương tác được, người chơi có thể tìm kiếm bí mật, thám hiểm và phát hiện ra các bí ẩn của vương quốc.

Trò chơi này hứa hẹn mang lại cho người chơi một trải nghiệm phiêu lưu đầy thách thức và kịch tính trong việc chiến đấu để giải cứu vương quốc khỏi sự ám ảnh của thế lực tà ác.

MỤC LỤC

DANH MỤC CÁC BẢNG VÀ HÌNH VẼ

Danh sách hình vẽ

Danh sách bảng

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

1.1. Cốt truyện

Trong thế giới ảo Rarel, vương quốc này đang đối diện với nguy cơ từ sự xâm chiếm của thế lực tà ác. Với công chúa bị bắt cóc và vương quốc đang chìm trong bóng tối, một người nông dân trẻ tuổi đã bước lên hành trình trở thành anh hùng và giải cứu vương quốc khỏi sự kiểm soát của thế lực đen tối.

1.2. Cách chơi

Người chơi sẽ đảm nhận vai trò của người anh hùng trẻ tuổi, điều khiển nhân vật qua các cấp độ khác nhau trong vương quốc Rarel. Họ sẽ phải đối mặt với các quái vật và Boss tàn ác, thu thập đồng vàng để mua trang bị và nâng cấp kỹ năng.

1.3. Các nhân vật và hệ thống

Các nhân vật chính, quái vật và Boss được thiết kế đa dạng, mỗi nhân vật đều có những đặc điểm và kỹ năng riêng.

Người chơi có thể nâng cấp nhân vật, sử dụng các kỹ năng và trang bị để đối phó với các thách thức trong game.

1.4. Cấp độ và môi trường

Game có nhiều cấp độ với môi trường đa dạng từ rừng sâu đến hang động bí ẩn, mang đến cho người chơi trải nghiệm mới mẻ và thú vị.

1.5. Sự cần thiết của đề tài

Games là một phần không thể thiếu trong thế giới giải trí hiện đại và có sức ảnh hưởng mạnh mẽ đến việc tương tác với công nghệ.

Đề tài này tạo ra một trải nghiệm giải trí hấp dẫn và kịch tính cho người chơi, đồng thời thúc đẩy sự sáng tạo và phát triển kỹ năng lập trình.

1.6. Yêu cầu của đề tài cần thực hiện

Thiết kế cốt truyện hấp dẫn và logic.

Phát triển hệ thống gameplay linh hoạt và thú vị.

Tạo ra các nhân vật và môi trường game đồng nhất và sâu sắc.

Xây dựng hệ thống nâng cấp và trang bị phong phú.

Đảm bảo sự tương tác và mức độ khó khăn phù hợp để tăng sự thú vị của trò chơi.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. ScriptableObject

2.1.1. Đối tượng dữ liệu Scriptable:

ScriptableObject là một lớp trong Unity cho phép bạn tạo ra các đối tượng dữ liệu tùy chỉnh mà không cần kế thừa từ MonoBehaviour. Điều này cho phép bạn tạo ra các loại dữ liệu như cài đặt, cấu hình, hoặc dữ liệu game mà có thể tái sử dụng và dễ dàng chỉnh sửa.

2.1.2. Tính linh hoạt và tái sử dụng:

ScriptableObject cho phép bạn tạo ra các loại dữ liệu phức tạp mà không cần tạo ra các instance riêng biệt cho mỗi đối tượng. Điều này giúp tăng tính linh hoạt và tái sử dụng của mã nguồn và dữ liệu trong game.

2.1.3. Khả năng lưu trữ và truy cập dữ liệu trong Inspector:

ScriptableObject có thể lưu trữ và hiển thị dữ liệu trong Inspector của Unity, cho phép bạn dễ dàng chỉnh sửa các giá trị của nó mà không cần phải sửa mã nguồn.

2.1.4. Dữ liệu có thể chia sẻ giữa các scene và instance:

ScriptableObject có thể chia sẻ dữ liệu giữa các scene và instance trong game, giúp bạn tránh tình trạng trùng lặp dữ liệu và giữ cho dữ liệu được duy trì một cách hiệu quả.

2.1.5. Tính năng Serialization và Asset Database:

ScriptableObject được hỗ trợ bởi hệ thống Serialization và Asset Database của Unity, cho phép bạn lưu trữ và quản lý ScriptableObject như các tài nguyên khác trong dự án của bạn.

2.1.6. Thực thi logic không cần tồn tại trong Scene:

ScriptableObject không cần tồn tại trong scene để thực thi logic. Điều này có nghĩa là bạn có thể sử dụng chúng để lưu trữ dữ liệu và logic mà không cần thực sự có một instance của đối tượng trong scene.

2.2. FSM (Finite State Machine)

2.2.1. Trạng thái (State):

Là các điều kiện hoặc tình trạng mà một hệ thống có thể tồn tại tại một thời điểm nhất định. Mỗi trạng thái biểu diễn cho một tình trạng cụ thể của hệ thống và có thể đáp ứng được một hoặc nhiều sự kiện.

2.2.2. Sự kiện (Event):

Là các tín hiệu hoặc điều kiện xảy ra trong hệ thống có thể gây ra sự chuyển đổi từ một trạng thái sang trạng thái khác. Mỗi sự kiện có thể kích hoạt hoặc gây ra sự thay đổi trong hành vi của hệ thống.

2.2.3. Bảng chuyển đổi trạng thái (Transition Table):

Là một bảng hoặc biểu đồ mô tả các quy tắc hoặc điều kiện để chuyển đổi từ một trạng thái sang trạng thái khác dựa trên sự kiện.

2.2.4. Hành vi (Behavior):

Là các hành động hoặc hoạt động mà hệ thống thực hiện khi ở trong một trạng thái cụ thể. Mỗi trạng thái có thể đi kèm với một hành vi riêng.

2.2.5. Quản lý trạng thái (State Management):

Là quá trình quản lý và điều khiển các trạng thái và chuyển đổi giữa chúng dựa trên các sự kiện xảy ra.

2.2.6. Lợi ích của FSM:

FSM giúp tổ chức và quản lý hành vi phức tạp thành các trạng thái và sự kiện dễ dàng hiểu và quản lý hơn. Nó cũng là một cách tiếp cận linh hoạt cho việc thực hiện logic điều khiển trong các ứng dụng và trò chơi.

2.3. Bleed Tree (Behavior tree)

2.3.1. Node (nút):

Node là các thành phần cơ bản của bleed tree, mỗi node thực hiện một chức năng cụ thể. Có ba loại node chính:

- Action Node: Node thực hiện một hành động cụ thể, như di chuyển, tấn công,

hoặc đứng yên.

- Composite Node: Node chứa một tập hợp các node con và quyết định cách thực thi chúng, có thể là một loạt các node tuần tự (sequence), một trong số các node con (selector), hoặc theo ngẫu nhiên (random selector).
- Decorator Node: Node thay đổi hoặc mở rộng hành vi của node con, như việc thêm điều kiện kiểm tra trước khi thực thi hành động.

2.3.2. Root Node (nút gốc):

Là node bắt đầu của bleed tree, mọi thứ bắt đầu từ đây. Root node thường là một composite node.

2.3.3. Leaf Node (nút lá):

Là node không có node con, thường là các action node. Trạng thái của node: Mỗi node có thể có một trạng thái, bao gồm trạng thái chờ, đang thực thi, và hoàn thành.

2.3.4. Quản lý trạng thái và chuyển đổi:

Bleed tree quản lý trạng thái và chuyển đổi giữa các node để đảm bảo rằng hành vi được thực hiện theo đúng thứ tự và logic.

2.3.5. Lợi ích của Bleed Tree:

Bleed tree cung cấp một cách tiếp cận cấu trúc hành vi có tổ chức và dễ hiểu. Nó giúp cho việc phát triển và điều chỉnh hành vi của nhân vật hoặc AI trở nên dễ dàng hơn và linh hoạt hơn.

2.4. PlayerPrefs

2.4.1. Lưu trữ dữ liệu cục bộ:

PlayerPrefs cho phép lưu trữ dữ liệu cục bộ trên thiết bị của người chơi, bao gồm các cài đặt, điểm số, mức độ hoàn thành, và các thông tin cá nhân khác mà người chơi muốn lưu lại.

2.4.2. Dữ liệu có thể lưu trữ dưới dạng key-value pairs:

PlayerPrefs sử dụng cấu trúc dữ liệu key-value pairs, trong đó mỗi giá trị được lưu trữ dưới dạng một key duy nhất và được truy cập thông qua key đó.

2.4.3. Dữ liệu lưu trữ có thể là các kiểu dữ liệu cơ bản:

PlayerPrefs hỗ trợ lưu trữ các kiểu dữ liệu cơ bản như số nguyên (integer), số thực (float), chuỗi (string), và boolean. Điều này cho phép lưu trữ và truy xuất các loại dữ liệu phổ biến một cách dễ dàng.

2.4.4. Dữ liệu lưu trữ trong tệp PlayerPrefs trong thiết bị:

PlayerPrefs lưu trữ dữ liệu trong các tệp cục bộ trên thiết bị của người chơi, nơi mà dữ liệu có thể được truy cập và cập nhật một cách nhanh chóng.

2.4.5. Khả năng chuyển đổi dữ liệu thành JSON:

PlayerPrefs cũng hỗ trợ chuyển đổi dữ liệu thành chuỗi JSON và lưu trữ dưới dạng chuỗi, cho phép lưu trữ và truy xuất dữ liệu phức tạp hơn.

2.4.6. Đơn giản và dễ sử dụng:

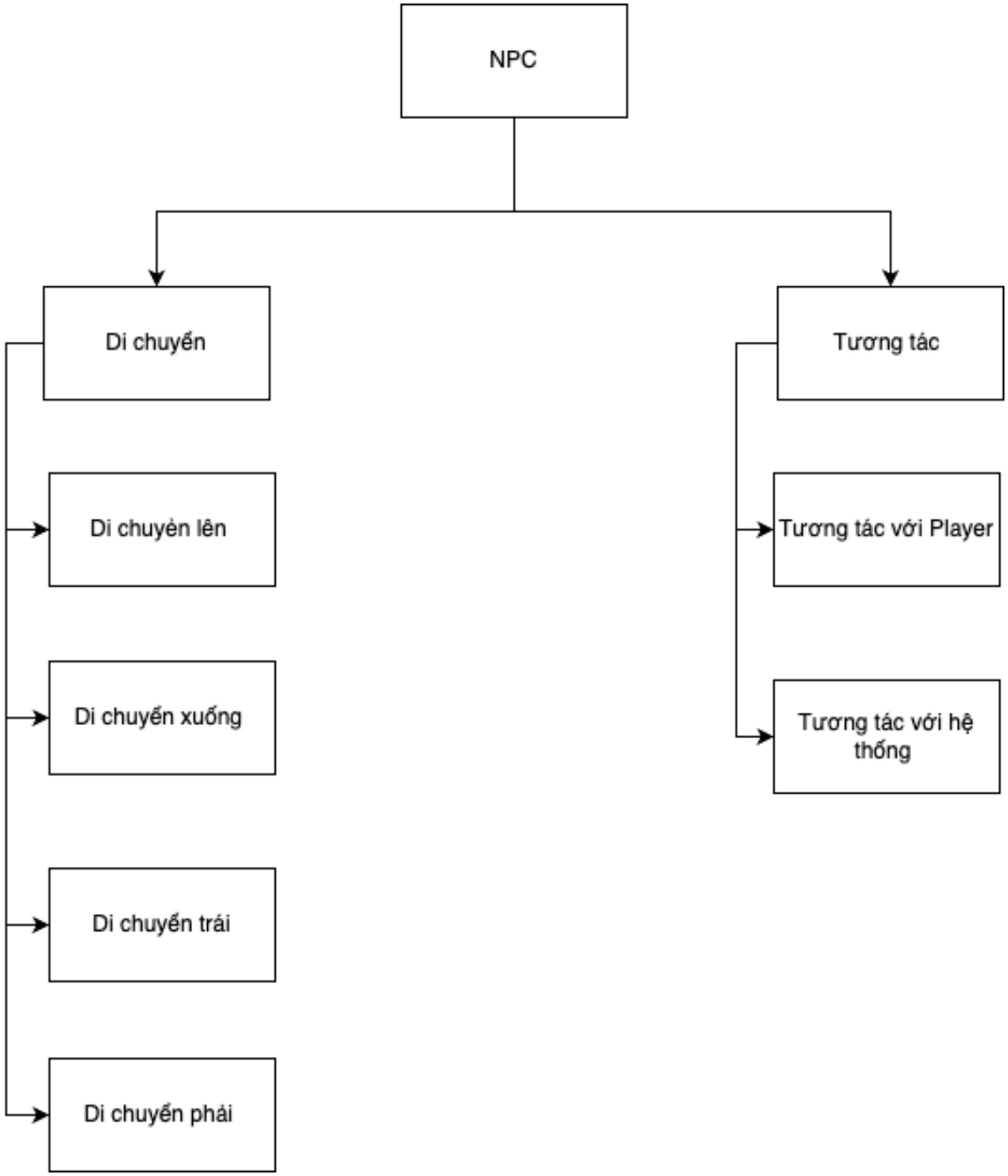
PlayerPrefs là một cơ chế lưu trữ dữ liệu đơn giản và dễ sử dụng trong Unity, giúp nhà phát triển tiết kiệm thời gian và công sức trong việc lưu trữ và truy xuất dữ liệu cục bộ của trò chơi.

CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ

3.1. Player

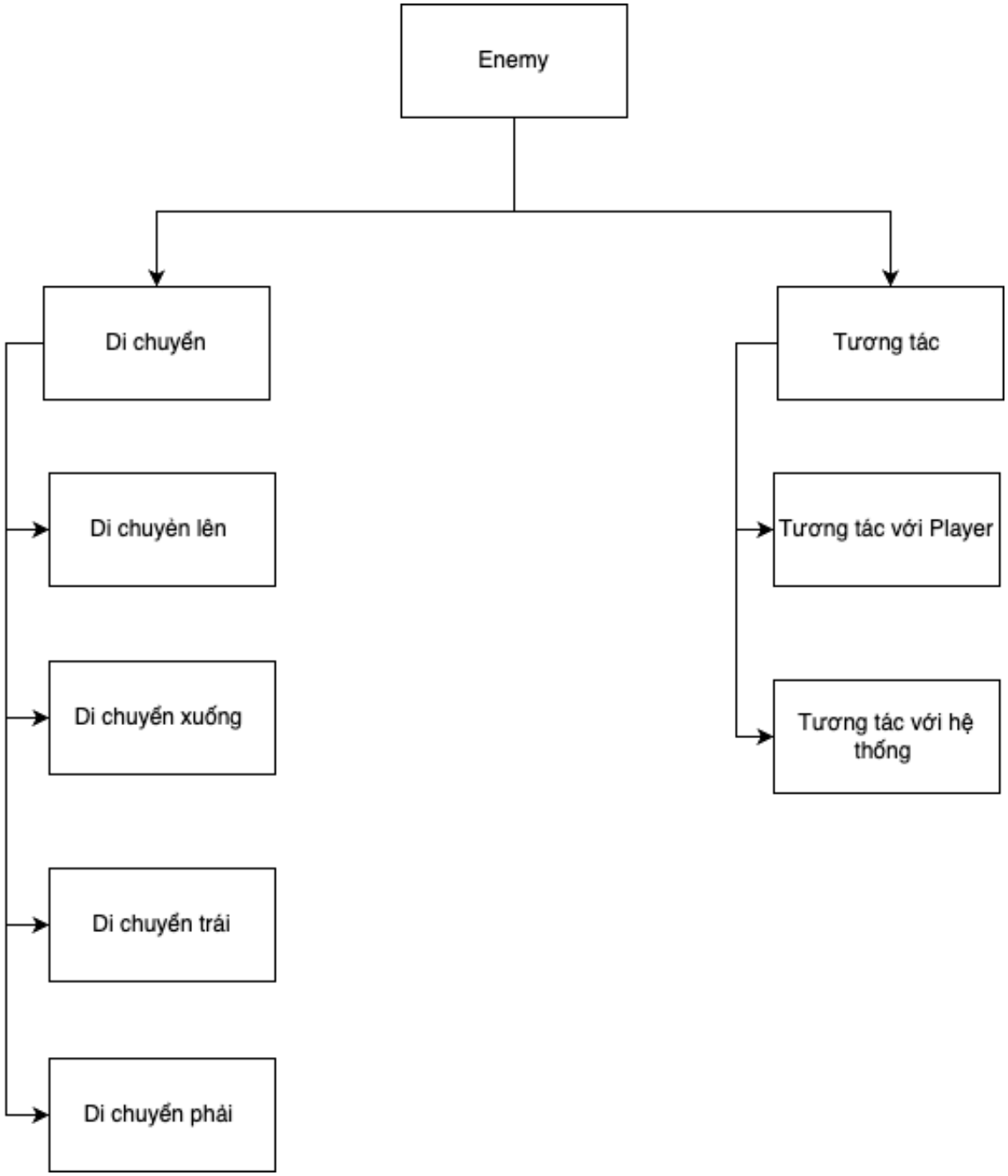
Hình 3.1: Sơ đồ phân rã chức năng của Player

3.2. NPC



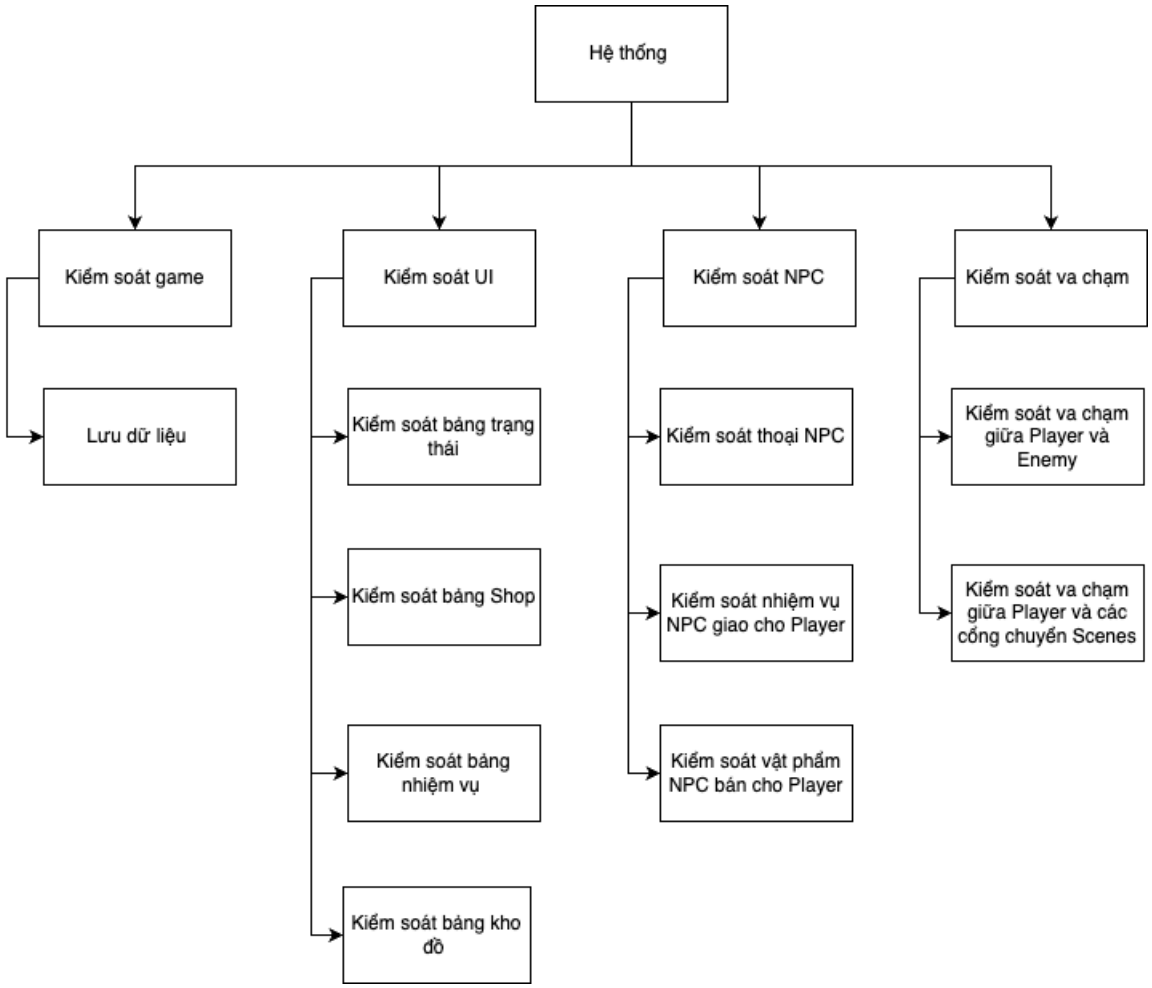
Hình 3.2: Sơ đồ phân rã chức năng của NPC

3.3. Enemy



Hình 3.3: Sơ đồ phân rã chức năng của Enemy

3.4. Hệ thống



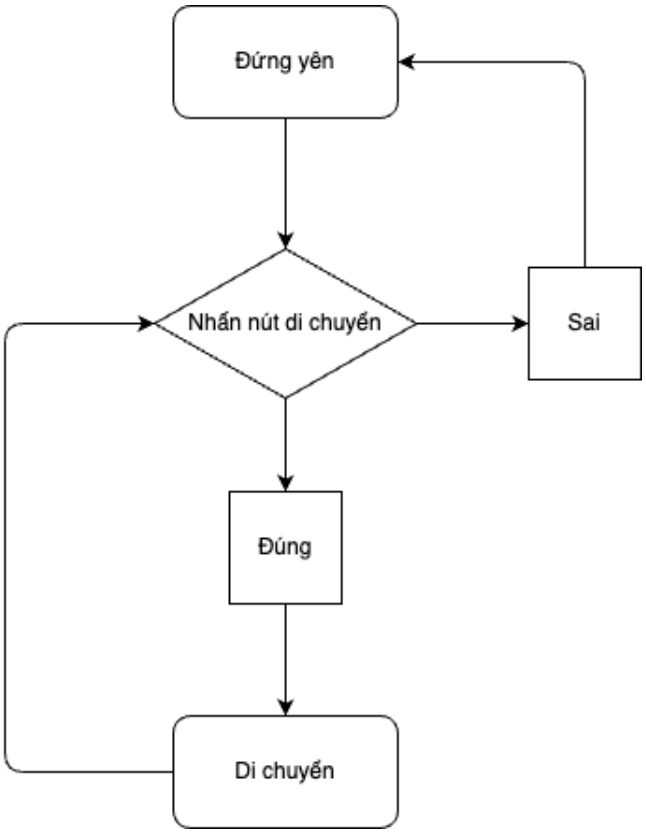
Hình 3.4: Sơ đồ phân rã chức năng của Hệ thống

CHƯƠNG 4: HIỆN THỰC HỆ THỐNG

4.1. Player

4.1.1. Di chuyển

Game sẽ liên tục kiểm tra xem người chơi có bấm các phím di chuyển không nếu có thì sẽ di chuyển. Di chuyển theo các hướng sẽ dựa vào thay đổi của x và y trong Vector2. Nếu giá trị x > 0 thì sẽ di chuyển sang hướng bên phải và ngược lại. Nếu giá trị y > 0 thì người chơi sẽ di chuyển lên trên và ngược lại. Cũng từ giá trị này ta sẽ đưa vào Bleed Tree để xét giá trị x,y và thiết lập các animations phù hợp với hướng di chuyển.

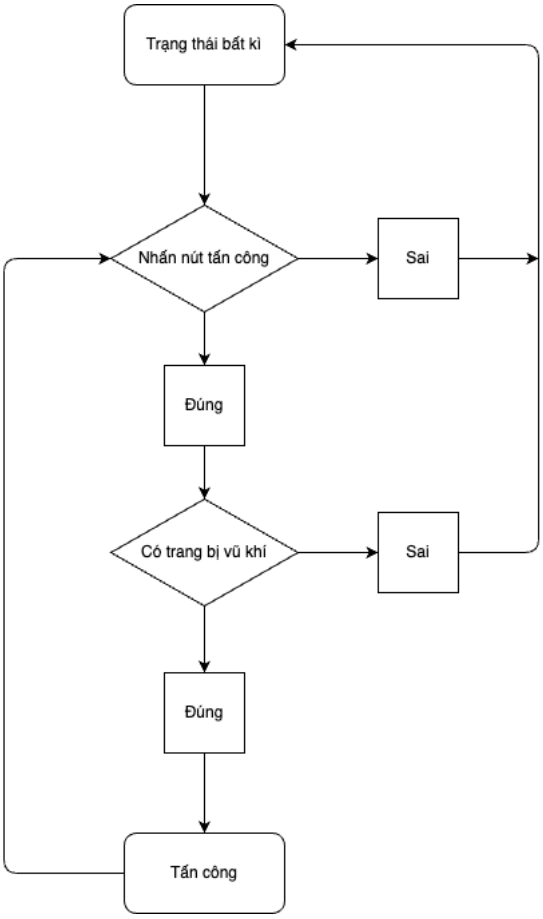


Hình 4.1: Workflow di chuyển của player

```
private void Update()
{
    ReadMoment();
}

private void FixedUpdate()
{
    Move();
}

private void Move()
{
    if (playerAttack.IsAttacking == true) return;
    if (player.Stats.Health <= 0) return;
    rb.MovePosition(rb.position + moveDirection * (speed * Time.fixedDeltaTime));
}

private void ReadMoment()
{
    if (playerAttack.IsAttacking == true) return;
    moveDirection = actions.Movement.Move.ReadValue<Vector2>().normalized;
    if (moveDirection == Vector2.zero)
    {
        playerAnimations.SetMoveBool(false);
        return;
    }
    playerAnimations.SetMoveBool(true);
    playerAnimations.SetMoveAnimation(moveDirection);
}
```

Hình 4.2: Code di chuyển của player

4.1.2. Tấn công



Hình 4.3: Workflow tấn công của player

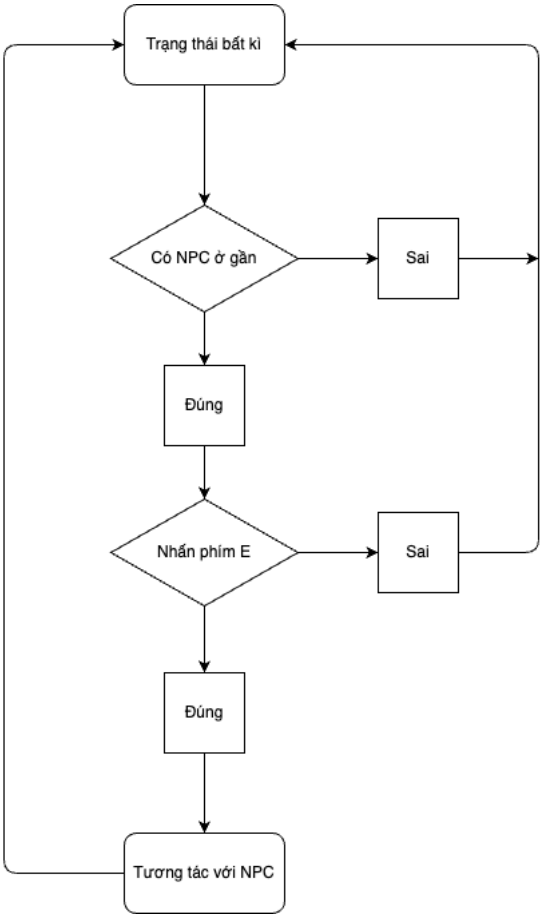
Khi người chơi đang ở trạng thái bất kì mà người chơi nhấn nút tấn công thì game sẽ kiểm tra xem là người chơi có đang trang bị vũ khí hay không nếu có thì sẽ kiểm tra tiếp lượng năng lượng hiện đang có của người chơi có đủ để thực hiện một hành động tấn công không nếu đã đạt đủ điều kiện sẽ được thực thi.

```
private IEnumerator IEAttack()
{
        if (currentAttackPosition == null) yield break;
        if (CurrentWeapon.WeaponType == WeaponType.Bow)
        {
            if (playerStamina.CurrentStamina < CurrentWeapon.RequiredStamina)
            {
                isAttacking = false;
                playerAnimations.SetAttackSwordAnimation(false);
                yield break;
            }
            BowAttack();
            isAttacking = true;
            playerAnimations.SetAttackBowAnimation(true);

            yield return new WaitForSeconds(0.5f);

            isAttacking = false;
            playerAnimations.SetAttackBowAnimation(false);
        }
        else
        {
            if (playerStamina.CurrentStamina < CurrentWeapon.RequiredStamina)
            {
                isAttacking = false;
                playerAnimations.SetAttackSwordAnimation(false);
                yield break;
            }
            MeleeAttack();
            isAttacking = true;
            playerAnimations.SetAttackSwordAnimation(true);

            yield return new WaitForSeconds(0.5f);

            isAttacking = false;
            playerAnimations.SetAttackSwordAnimation(false);
        }

}
```

Hình 4.4: Code tấn công của player

Ở hàm tấn công này chúng em sẽ sử dụng IEnumerator vì em muốn kiểm soát được animations và projectile là mũi tên được trùng khớp với nhau nên sẽ có khoản thời gian nghỉ là khoản 0.5s để cho 2 thứ được khớp với nhau.

4.1.3. Tương tác



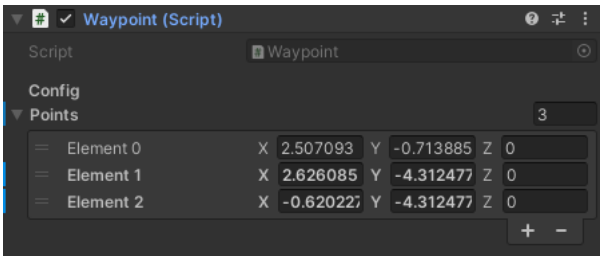
Hình 4.5: Workflow tương tác với NPC của player

NPC sẽ có một circle collider được đặt là trigger nó được kích hoạt khi có collider của người chơi nằm trong phạm vi. Khi đó nút chức năng tương tác sẽ được hiển thị để người chơi có thể tương tác được với NPC.

Người chơi cũng có thể tương tác hệ thống với các button chức năng trên màn hình game. Khi nhấn vào thì người chơi sẽ tăng chỉ số sử dụng vật phẩm hay là kiểm tra nhiệm vụ của bản thân.

4.2. NPC

4.2.1. Di chuyển



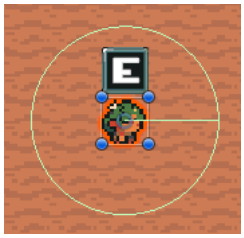
Hình 4.6: Waypoint

Khác với người chơi thì NPC sẽ sử dụng các waypoint để di chuyển và các điểm di chuyển sẽ được cố định. NPC sẽ đọc giá trị của các waypoint trước và waypoint tiếp theo cần di chuyển để trả về các giá trị x y trong Vector2 để xác định được hướng cần di chuyển.

```
private void Update()
{
    Vector3 nextPos = waypoint.GetPosition(currentPointIndex);
    UpdateMoveValues(nextPos);
    transform.position = Vector3.MoveTowards(transform.position,
        nextPos, moveSpeed * Time.deltaTime);
    if (Vector3.Distance(transform.position, nextPos) <= 0.2)
    {
        previousPos = nextPos;
        currentPointIndex = (currentPointIndex + 1) % waypoint.Points.Length;
    }
}

private void UpdateMoveValues(Vector3 nextPos)
{
    Vector2 dir = Vector2.zero;
    if (previousPos.x < nextPos.x) dir = new Vector2(1f, 0f);
    if (previousPos.x > nextPos.x) dir = new Vector2(-1f, 0f);
    if (previousPos.y < nextPos.y) dir = new Vector2(0f, 1f);
    if (previousPos.y > nextPos.y) dir = new Vector2(0f, -1f);
    animator.SetFloat(moveX, dir.x);
    animator.SetFloat(moveY, dir.y);
}
```

Hình 4.7: Code di chuyển của NPC

4.2.2. Tương tác



Hình 4.8: NPC và collider trigger

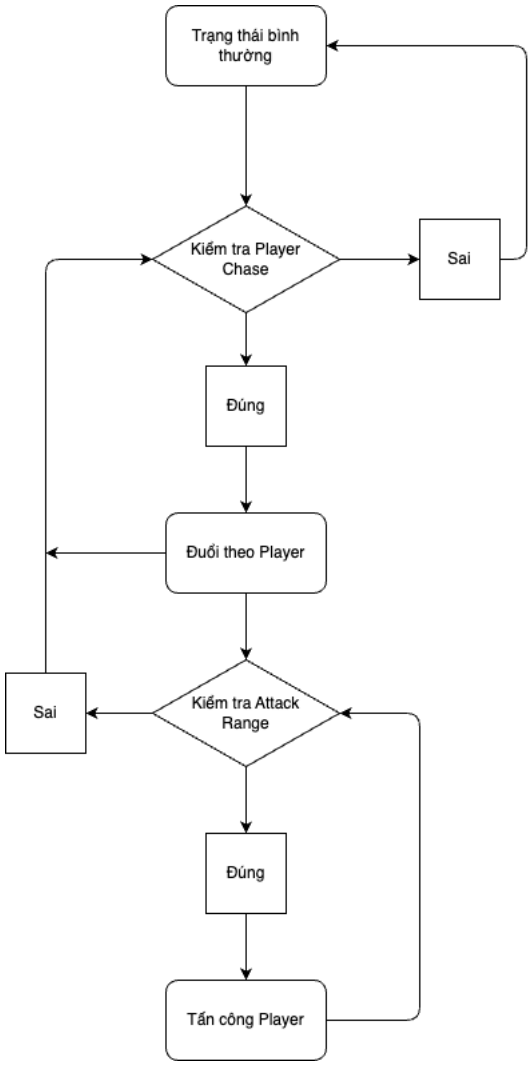
NPC sẽ có một collider trigger để kiểm tra xem người chơi có đang ở trong khu vực hoạt động của trigger hay không nếu có thì sẽ hiển thị phím chức năng để người chơi tương tác.

Hình 4.9: NPC Shop

Shop và nhiệm vụ sẽ giống nhau ban đầu các Pannel sẽ không có vật phẩm hay nhiệm vụ nào cả khi game bắt đầu và người chơi tương tác với NPC thì lúc đó hệ thống sẽ kiểm tra NPC đó đang giữ chức năng gì là shop hay là nhiệm vụ thông qua Sciptableobject sau đó sẽ cung cấp các vật phẩm hoặc nhiệm vụ để hiện thì ra với người chơi.

4.3. Enemy

4.3.1. Di chuyển và tương tác



Hình 4.10: Flow sự kiện Enemy

Enemy sẽ di chuyển giống với NPC bằng các waypoint. Enemy sẽ có 2 vùng sự kiện là chase và attack range khi Player đi vào các vùng sự kiện nào thì ứng với nó thì sự kiện đó sẽ được diễn ra và khi Player thoát khỏi tất cả các vùng điều kiện thì sẽ trở lại trạng thái bình thường.

```
private bool DetectPlayer()
{
    Collider2D playerCollider =
        Physics2D.OverlapCircle(enemyAI.transform.position,
            range, playerMask);
    if (playerCollider != null)
    {
        enemyAI.Player = playerCollider.transform;
        return true;
    }

    enemyAI.Player = null;
    return false;
}

private void OnDrawGizmosSelected()
{
    Gizmos.color = Color.red;
    Gizmos.DrawWireSphere(transform.position, range);
}
}
```

Hình 4.11: Code kiểm tra sự kiện của Enemy

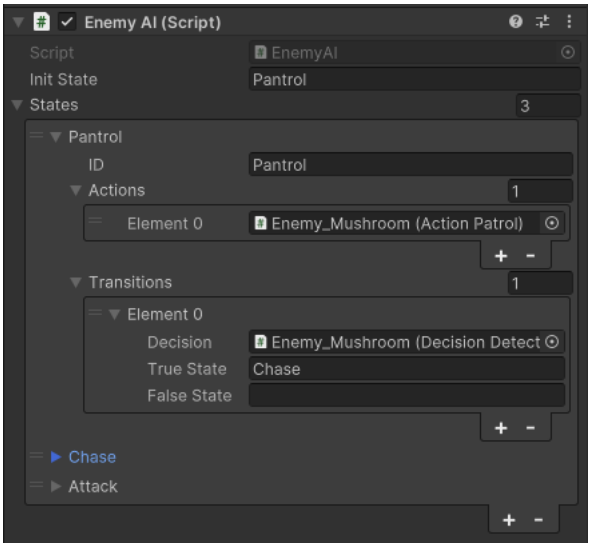
Khác với NPC thì Collider của Enemy sẽ được tạo ra trong code để kiểm tra việc người chơi có đi vào vùng điều kiện hay không. Điều này sẽ tránh được việc tạo quá nhiều component Collider trong gameobject. Các collider sự kiện sẽ được vẽ bằng DrawGizmos để tiện việc kiểm tra.

CHƯƠNG 5: KẾT QUẢ ĐẠT ĐƯỢC

Hình 5.1: NPC bình thường tương tác với người chơi

Hình 5.2: NPC đặc biệt tương tác với người chơi

Hình 5.3: Animations và projectile đồng bộ với nhau



Hình 5.4: FSM của Enemy

Hình 5.5: Các collider sự kiện được tạo ra và vẽ nó bằng DrawGizmos

CHƯƠNG 6: KẾT LUẬN

6.1. Ưu điểm

Cốt truyện: Cốt truyện của game là một điểm mạnh, với một thế giới ảo phong phú và nhân vật đầy tính cách, tạo ra một trải nghiệm phiêu lưu đầy kịch tính cho người chơi.

Gameplay đa dạng: Việc tích hợp nhiều loại hình gameplay như di chuyển, chiến đấu, thu thập đồng vàng và nâng cấp kỹ năng tạo ra sự đa dạng và thú vị cho người chơi.

6.2. Khuyết điểm

Cốt truyện chưa có chiều sâu còn sơ sài.

Thiếu sự tương tác: Game có thể cải thiện bằng cách thêm nhiều tính năng tương tác hơn, như các nhiệm vụ phụ, các khu vực có thể tương tác và các sự kiện ngẫu nhiên.

Điều chỉnh cân bằng: Cần điều chỉnh cân bằng gameplay, bao gồm việc điều chỉnh độ khó của các trận đấu với Boss và việc điều chỉnh sự hiện diện của các vật phẩm và trang bị.

6.3. Đã làm được

Xây dựng cốt truyện và thế giới game.

Phát triển gameplay đa dạng và thú vị.

Tích hợp đồ họa và âm nhạc chất lượng.

6.4. Chưa làm được làm được

Xây dựng cốt truyện và thế giới game phong phú có chiều sâu

Tối ưu hóa trải nghiệm người chơi thông qua việc tương tác và cân bằng gameplay.

Kỹ thuật tạo ra môi trường tương tác và sự kiện ngẫu nhiên để tăng tính tương tác và tái chơi của game.

6.5. Hướng phát triển tương lai cho game

Mở rộng tính tương tác: Tăng cường tính tương tác bằng cách thêm nhiều tính năng mới như các nhiệm vụ phụ, NPC tương tác và các sự kiện động.

Cải thiện cân bằng gameplay: Điều chỉnh độ khó của trò chơi, cân nhắc về việc thêm các phần thưởng và thách thức mới để tạo ra một trải nghiệm cân bằng và hấp dẫn hơn.

Tối ưu hóa hiệu suất: Tối ưu hóa game để đảm bảo hiệu suất tốt nhất trên các thiết bị khác nhau, từ điện thoại di động đến máy tính cá nhân.

Mở rộng thế giới và nội dung: Thêm nhiều cấp độ, khu vực và nhiệm vụ mới để mở rộng thế giới game và tạo ra nhiều cơ hội khám phá cho người chơi.

TÀI LIỆU THAM KHẢO

[1] Docs.unity.com. "Unity Documentation."

Available: https://docs.unity.com .