

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN
MÔN THỰC TẬP CƠ SỞ

Đề tài : xây dựng game 2D cơ bản

Giảng viên hướng dẫn : Bùi Văn Kiên

Họ và tên sinh viên : Trần Thanh Thảo

Mã sinh viên : B22DCCN804

Nhóm : 25

Hà Nội – 2025

Mục Lục

Mục Lục	1
Danh Mục Hình Ảnh	3
Danh mục bảng.....	4
Lời mở đầu	5
CHƯƠNG I: GIỚI THIỆU ĐỒ ÁN VÀ CÁC CÔNG NGHỆ SỬ DỤNG.....	6
1.1 Giới thiệu đồ án	6
1.2 Các công nghệ sử dụng	6
1.2.1 Unity 2D:	6
1.2.2 C#:.....	7
1.2.3 Unity UI (Canvas):	7
1.2.4 PlayerPrefs:	7
1.2.5 Git và GitHub:	8
1.2.6 Các thư viện Unity khác:	8
CHƯƠNG II : THIẾT KẾ CỐT TRUYỆN VÀ LÊN Ý TƯỞNG CHO TRÒ CHƠI.....	8
2.1 Lên ý tưởng chung	8
2.1.1 Cốt truyện	8
2.1.2 Thể loại	9
2.1.3 Số lượng người chơi	9
2.1.4 Logic cơ bản	9
2.1.5 Tính năng khác	10
2.2 Thiết kế và lên ý tưởng cho các đối tượng trong trò chơi.....	10
CHƯƠNG III: PHÂN TÍCH VÀ THIẾT KẾ TRÒ CHƠI.....	12
3.1 Phân tích hệ thống.....	12
3.1.1 Mô tả dự án.....	12
3.1.2 Mô tả hệ thống bằng UML	14
3.1.3 Kịch bản chuẩn và ngoại lệ	16
3.1.4. Usecase tổng quan	19
3.2 Thiết kế hệ thống trò chơi	21

3.2.1 Sơ đồ lớp thực thể hệ thống.....	21
3.2.2 Sơ đồ lớp module Thu thập item khi quái chết	27
3.2.3 Sơ đồ lớp module Tấn công kẻ địch pha thiết kế.	28
3.2.3 Sơ đồ lớp module Gán phím tương tác cho nhân vật.	29

Danh Mục Hình Ảnh

Hình 1: Sơ đồ usecase tổng quan	19
Hình 2: Biểu đồ lớp thực thể tổng quan của trò chơi	21
Hình 3: Biểu đồ lớp module Thu thập item khi quái chết.....	27
Hình 4: Biểu đồ lớp module Tấn công kẻ địch	28
Hình 5: Biểu đồ lớp module Gán phím tương tác cho nhân vật	29
Hình 6: Màn hình Menu chính khi vào game	31
Hình 7: Màn hình cài đặt của game.....	31
Hình 8: Màn hình giao diện khi bắt đầu vào chơi game	32
Hình 9 : Màn hình khi chiến đấu kẻ địch (quái nắm).....	32
Hình 10: Màn hình khi quái bị hạ xuất hiện hiệu ứng nổ cùng vật phẩm thưởng	32
Hình 11: Màn hình các loại bẫy xuất hiện trong game	33
Hình 12: Màn hình khi gặp Boss game	34
Hình 13: Màn hình boss triệu hồi minion tấn công.....	34
Hình 14: Màn hình nhận được chìa khóa sau khi boss bị hạ	36
Hình 15: Màn hình khi thua game.....	36
Hình 16: Màn hình hiển thị thắng khi thu được chìa khóa vàng.....	37

Danh mục bảng

Bảng 1: Bảng ý tưởng các nhân vật trong trò chơi	10
Bảng 2: Bảng lên ý tưởng cho các đối tượng trong trò chơi	11
Bảng 3: Bảng từ khóa	14
Bảng 4 : Bảng Scenario Thu thập item khi quái chết	16
Bảng 5 : Bảng Scenario chức năng tấn công kẻ địch	17
Bảng 6 : Bảng Scenario chức năng Gán phím tương tác cho nhân vật	18
Bảng 7: Bảng mô tả tổng quan các lớp liên quan	21
Bảng 8: Bảng mô tả lớp SoundManager	22
Bảng 9: Bảng mô tả lớp Menu	22
Bảng 10: Bảng mô tả lớp Settings	22
Bảng 11: Bảng mô tả lớp Player	23
Bảng 12: Bảng mô tả lớp Statistics	23
Bảng 13: Bảng mô tả lớp GameplayUI	24
Bảng 14: Bảng mô tả lớp Enemy	24
Bảng 15: Bảng mô tả lớp Minion	25
Bảng 16: Bảng mô tả lớp Boss	25
Bảng 17: Bảng mô tả lớp MushFoe	25
Bảng 18: Bảng mô tả lớp Item	25
Bảng 19: Bảng mô tả lớp CameraShake	26
Bảng 20: Bảng mô tả lớp KeyEventManager	26
Bảng 21: Bảng mô tả lớp Trap	26
Bảng 22: Bảng mô tả các phương thức của lớp module Thu thập item khi quái chết	27
Bảng 23: Bảng mô tả các phương thức của lớp module Tấn công địch	28
Bảng 24: Bảng mô tả các phương thức trong lớp module Gán phím	29

Lời mở đầu

Hiện nay, sự gia tăng và phát triển của các thiết bị di động đã mở ra một thị trường game di động rộng lớn, ngành game đang ngày càng phát triển và trở thành xu hướng. Trong bối cảnh đó, em quyết định chọn Unity 2D làm nền tảng phát triển game của mình bởi những ưu điểm vượt trội mà nó mang lại.

Unity 2D không chỉ hỗ trợ tạo ra các trò chơi 2D một cách dễ dàng và hiệu quả với giao diện thân thiện, mà còn cung cấp một hệ sinh thái phong phú với các công cụ mạnh mẽ như hệ thống vật lý, quản lý animation và khả năng tích hợp đa nền tảng. Đặc biệt, Unity 2D cho phép xuất game sang nhiều thiết bị di động khác nhau như Android và iOS một cách nhanh chóng, giúp tiết kiệm thời gian và chi phí phát triển.

Ngoài ra, cộng đồng phát triển đông đảo cùng tài liệu phong phú của Unity là nguồn hỗ trợ đắc lực, giúp em rất nhiều trong việc học hỏi và tối ưu hóa quá trình phát triển game của mình. Việc chọn Unity 2D không chỉ phù hợp với mục tiêu xây dựng một game 2D đơn giản, dễ tiếp cận mà còn mở ra cơ hội phát triển các dự án phức tạp hơn trong tương lai.

CHƯƠNG I: GIỚI THIỆU ĐỒ ÁN VÀ CÁC CÔNG NGHỆ SỬ DỤNG

1.1 Giới thiệu đồ án

- Đồ án của là dự án phát triển game 2D đơn giản nhưng tương đối đầy đủ tính năng, được xây dựng nhằm mục đích học tập và thực hành các kỹ năng lập trình game trên nền tảng Unity.
- Game tập trung vào việc tạo ra một trải nghiệm chơi game mượt mà với các yếu tố chính như điều khiển nhân vật, hệ thống tấn công, thu thập vật phẩm và quản lý phẩm tùy chỉnh. Người chơi có thể điều khiển nhân vật di chuyển (trái, phải, nhảy), tấn công kẻ thù, thu thập đồng xu, tránh bẫy, và hoàn thành màn chơi khi lấy được chìa khóa.
- Điểm nổi bật của đồ án là hệ thống gán phím (keybinding) linh hoạt, cho phép người chơi tùy chỉnh phím điều khiển thông qua giao diện menu, đồng thời lưu trữ các thiết lập này để sử dụng trong các lần chơi sau. Với các Scene như SetButton (gán phím), Menu (truy cập các tùy chọn), và Level1 (màn chơi chính), em đã cố gắng phát triển dự án thành một quy trình phát triển game tương đối hoàn chỉnh từ thiết kế giao diện, quản lý trạng thái game, đến xử lý logic nhân vật.

1.2 Các công nghệ sử dụng

1.2.1 Unity 2D:

- **Vai trò:** Là nền tảng chính để phát triển game, cung cấp môi trường thiết kế và lập trình 2D mạnh mẽ.
- **Lý do chọn:** Unity 2D được chọn vì giao diện thân thiện, hỗ trợ tích hợp đa nền tảng (Android, iOS), và hệ sinh thái phong phú với các công cụ như hệ thống vật lý 2D (Rigidbody2D, Collider2D), quản lý animation (Animator), và khả năng xuất game dễ dàng.
- **Ứng dụng trong đồ án:**
 - o Sử dụng Rigidbody2D để xử lý di chuyển và nhảy của nhân vật (Player.cs).

- Sử dụng Animator để điều khiển các trạng thái animation như đi bộ (Walk), nhảy (Jump), và tấn công (Attack1, Attack2, Attack3).
- Tích hợp Physics2D.OverlapCircle để phát hiện kẻ thù trong phạm vi tấn công.

1.2.2 C#:

- **Vai trò:** Là ngôn ngữ lập trình chính để viết logic game trong Unity.
- **Ứng dụng trong đồ án:**
 - Quản lý toàn cục với GameManager.cs (singleton pattern) để lưu trữ và truy cập phím điều khiển (left, right, jump, attack).
 - Xử lý đầu vào người chơi (Input.GetKey, Input.GetKeyDown) trong Player.cs và MenuScript.cs.
 - Lưu trữ dữ liệu phím với PlayerPrefs để giữ thiết lập người chơi qua các lần chơi.

1.2.3 Unity UI (Canvas):

- **Vai trò:** Được sử dụng để xây dựng giao diện người dùng, bao gồm menu gán phím và hiển thị thông tin game.
- **Ứng dụng trong đồ án:**
 - Sử dụng Canvas trong Scene SetButton để tạo giao diện gán phím với các nút (Button) và văn bản (TextMeshProUGUI hoặc Text).
 - Trong Player.cs, hiển thị số đồng xu (currentCoin) và máu tối đa (maxHealth) qua Text components.
 - Trong MenuScript.cs, tạo menu gán phím với các nút (leftKey, rightKey, v.v.) và hiển thị phím hiện tại qua Text.
 - Sử dụng TextMeshProUGUI trong KeybindingManager.cs để hiển thị và cập nhật phím điều khiển (leftText, rightText, v.v.) trên giao diện.
 - Cung cấp khả năng hiển thị văn bản sắc nét và dễ tùy chỉnh hơn so với Text thông thường.

1.2.4 PlayerPrefs:

- **Vai trò:** Là công cụ lưu trữ dữ liệu đơn giản của Unity, dùng để lưu và tải các thiết lập của người chơi.
- **Ứng dụng trong đồ án:**
 - o Trong GameManager.cs, lưu trữ và tải các phím tùy chỉnh (leftKey, rightKey, jumpKey, attackKey) từ PlayerPrefs.
 - o Trong KeybindingManager.cs và MenuScript.cs, lưu các phím được gán vào PlayerPrefs để sử dụng trong các lần chơi sau.

1.2.5 Git và GitHub:

- **Vai trò:** Quản lý mã nguồn và triển khai dự án.
- **Ứng dụng trong đồ án:**
 - o Sử dụng Git để quản lý các thay đổi mã nguồn trong dự án (thư mục /u/Kì 2 năm 3/ttcs/game2d/Assets).
 - o Đẩy code lên GitHub (<https://github.com/thankthao34/ttcs.git>) để lưu trữ và chia sẻ dự án.

1.2.6 Các thư viện Unity khác:

- **Unity.Mathematics:** Sử dụng trong Player.cs để xử lý tính toán (như Mathf.Abs để kiểm tra di chuyển và điều khiển animation).
- **UnityEngine.SceneManagement:** Sử dụng trong KeybindingManager.cs để chuyển đổi Scene (SceneManager.LoadScene("Menu")) sau khi lưu phím.

CHƯƠNG II : THIẾT KẾ CỐT TRUYỆN VÀ LÊN Ý TƯỞNG CHO TRÒ CHƠI

2.1 Lên ý tưởng chung

2.1.1 Cốt truyện

Dự án của em lấy bối cảnh một vương quốc cổ xưa bị nguyên rủa, nơi nhân vật chính - một chiến binh dũng cảm - phải vượt qua hàng loạt thử thách để giải cứu vương quốc. Cốt truyện xoay quanh hành trình của nhân vật đi qua các khu rừng bí ẩn, vượt qua các cạm bẫy,

chiến đấu với các kẻ thù như quái vật và thủ lĩnh “Undead”(boss game), thu các vật phẩm để tăng cường sức mạnh cả vật chất lẫn tinh thần, và tìm kiếm chìa khóa vàng để phá bỏ lời nguyền. Mỗi màn chơi là một phần của cuộc hành trình, với những bẫy nguy hiểm và kẻ thù ngày càng mạnh mẽ. Mục tiêu cuối cùng là đối mặt với tên thủ lĩnh “Undead” để giành lại hòa bình cho vương quốc, mang lại cảm giác phiêu lưu và kịch tính cho người chơi.

2.1.2 Thể loại

Game thuộc thể loại platformer 2D kết hợp yếu tố action-adventure. Người chơi sẽ trải nghiệm lối chơi nhảy qua các nền tảng, chiến đấu với kẻ thù, và khám phá môi trường. Sự kết hợp này tạo nên một trải nghiệm vừa thử thách về kỹ năng điều khiển, vừa hấp dẫn với các yếu tố khám phá và chiến đấu.

2.1.3 Số lượng người chơi

Game được thiết kế cho một người chơi (single-player). Mỗi người chơi điều khiển nhân vật chính thông qua các phím tùy chỉnh, tập trung vào trải nghiệm cá nhân với các thử thách độc lập trong từng màn chơi. Tuy nhiên, có tiềm năng mở rộng thành chế độ nhiều người chơi trong tương lai nếu dự án được phát triển thêm.

2.1.4 Logic cơ bản

Logic cơ bản của game được xây dựng dựa trên các thành phần chính sau:

- **Di chuyển và nhảy:** Nhân vật sử dụng phím tùy chỉnh (lưu trong GameManager.GM) để di chuyển trái/phải và nhảy qua các nền tảng, được xử lý bằng RigidBody2D và Input.GetKey trong Player.cs.
- **Tấn công:** Người chơi tấn công kẻ thù bằng các animation ngẫu nhiên (Attack1, Attack2, Attack3) khi nhấn phím tấn công, sử dụng Physics2D.OverlapCircle để phát hiện va chạm với kẻ thù trong PlayerAttack.
- **Tương tác với môi trường:** Nhân vật thu thập đồng xu, tránh bẫy, và hoàn thành màn chơi khi lấy chìa khóa.
- **Quản lý máu:** Hệ thống máu giảm khi va chạm với bẫy hoặc kẻ thù, dẫn đến trạng thái game over khi maxHealth \leq 0.

- **Menu và gán phím:** Giao diện menu cho phép người chơi bật/tắt menu bằng phím Escape, gán phím mới, và lưu vào PlayerPrefs để duy trì thiết lập.

2.1.5 Tính năng khác

- **Hiệu ứng đặc biệt:** Sử dụng hiệu ứng nổ (explosionPrefab) khi nhân vật chết và văn bản nổi (floatingtextprefab) khi gây sát thương kẻ thù, tăng tính trực quan trong Player.cs.
- **Rung màn hình (Camera Shake):** Tích hợp hiệu ứng rung màn hình khi nhân vật nhận sát thương hoặc chết, tạo cảm giác chân thực hơn.
- **Âm thanh:** Tích hợp âm thanh nhảy thông qua FindObjectOfType<Sound>(), tăng trải nghiệm đa phương tiện.
- **Giao diện người dùng (UI):** Hiển thị số đồng xu (currentCointext) và máu tối đa (maxHealthText) trên màn hình, cùng với menu gán phím tùy chỉnh trong Scene SetButton.
- **Tùy chỉnh phím:** Hỗ trợ người chơi thay đổi phím điều khiển (left, right, jump, attack) qua giao diện menu, lưu trữ bằng PlayerPrefs, và đồng bộ với GameManager.

2.2 Thiết kế và lên ý tưởng cho các đối tượng trong trò chơi

Bảng 1: Bảng ý tưởng các nhân vật trong trò chơi

Stt	Tên	Máu	Tốc chạy	Mô tả kỹ năng
1	Chiến binh	10	7	<ul style="list-style-type: none">- Đây là nhân vật chính mà người chơi điều khiển. Nhân vật cầm một thanh kiếm, đây cũng là vật gây sát thương chính lên các kẻ thù.- Có khả năng di chuyển trái/phải, nhảy qua các nền tảng, và tấn công kẻ thù với 3 kiểu kỹ năng tấn công ngẫu nhiên (Attack1, Attack2, Attack3). Tấn công gây 1 sát thương lên kẻ thù trong phạm vi (bán kính 1.5). Có thể thu thập đồng xu để tăng điểm và tìm chìa khóa để hoàn thành màn chơi.- Ba kỹ năng của nhân vật bao gồm :

				Kỹ năng	Mô tả hành động	Sát thương vật lý	Thời gian hồi chiêu
				Attack1	Chém kiếm từ trên xuống	1	Ngẫu nhiên
				Attack2	Vung kiếm từ dưới chém lên	1	Ngẫu nhiên
				Attack3	Đâm kiếm trực diện	1	Ngẫu nhiên

Bảng 2: Bảng lên ý tưởng cho các đối tượng trong trò chơi

Stt	Tên	Máu	Tốc chạy	Mô tả	Sát thương
1	Quái vật rừng	3	4	<p>Một loại kẻ thù cơ bản xuất hiện trong khu rừng. Di chuyển qua lại giới hạn trong 1 khu vực nhất định, tấn công nhân vật chính khi tiếp cận, gây 1 sát thương mỗi lần va chạm.</p> <p>Kỹ năng: Tăng tốc chạy ngắn hạn khi phát hiện người chơi ở gần (trong bán kính 5).</p>	1
2	Tiểu quái/minion	2	5	<p>Một loại kẻ thù nhỏ xuất hiện thành nhóm. Di chuyển nhanh nhưng máu thấp, dễ bị tiêu diệt.</p> <p>Kỹ năng: Tấn công bằng cách lao thẳng vào người chơi</p> <p>Có thể xuất hiện liên tục mỗi khi quái boss aka "Undead" triệu hồi trong màn chơi.</p>	1

3	Thủ lĩnh Undead	30	3	Trùm cuối (Undead_Boss) của màn chơi, bảo vệ chìa khóa vàng. Di chuyển chậm nhưng có sức mạnh lớn. Kỹ năng: Triệu hồi tiểu quái (1 con mỗi lần) và tấn công bằng sóng năng lượng.	2
---	--------------------	----	---	--	---

CHƯƠNG III: PHÂN TÍCH VÀ THIẾT KẾ TRÒ CHƠI

3.1 Phân tích hệ thống

3.1.1 Mô tả dự án

a. Xác định phạm vi dự án

Dự án em xây dựng là một game 2D thuộc thể loại platformer kết hợp yếu tố hành động phiêu lưu (action-adventure), được phát triển trên nền tảng Unity 2D. Phạm vi của dự án tập trung vào việc xây dựng một trò chơi đơn giản nhưng đầy đủ các tính năng cơ bản, phù hợp với mục đích học tập và thực hành kỹ năng lập trình game. Cụ thể:

- Phạm vi chức năng:
 - Gameplay: Bao gồm điều khiển nhân vật, tương tác với môi trường, và chiến đấu với các loại kẻ thù .
 - Giao diện người dùng (UI): Xây dựng giao diện hiển thị thông tin (số đồng xu, máu), menu gán phím, và các màn hình kết thúc.
 - Quản lý dữ liệu: Lưu trữ và tải các thiết lập phím điều khiển của người chơi thông qua PlayerPrefs.
 - Hiệu ứng và âm thanh: Tích hợp hiệu ứng trực quan (nổ, văn bản nổi, rung màn hình) và âm thanh để tăng trải nghiệm.
- Phạm vi kỹ thuật:
 - Sử dụng Unity 2D và ngôn ngữ lập trình C#.
 - Tích hợp các thư viện của Unity như UnityEngine và TextMeshPro để xây dựng giao diện và quản lý Scene.
 - Quản lý mã nguồn bằng Git và triển khai trên Github.

- Đối tượng người chơi: Game hướng đến những người chơi yêu thích thể loại platformer 2D, đặc biệt là người chơi mới hoặc sinh viên học lập trình game muốn trải nghiệm một trò chơi đơn giản nhưng đầy đủ tính năng.

b. Tính năng trò chơi chi tiết

- Chức năng **khởi động và hiển thị màn chơi**: Người chơi bắt đầu từ Scene Menu, màn hình hiện 3 nút chính, 1 nút “Play” người chơi bắt đầu điều khiển nhân vật chính để khám phá màn chơi, tìm kiếm chìa khóa và hoàn thành nhiệm vụ., 1 nút “Exit” thoát khỏi game, 1 nút để tùy chỉnh cài đặt của game .
- Chức năng **tùy chỉnh cài đặt của game**: Người chơi bắt đầu từ Scene SetButton hoặc mở menu trong game bằng phím Escape -> Giao diện gán phím hiện lên với các nút (leftKey, rightKey, jumpKey, attackKey), 1 slider thay đổi âm thanh nền của game và nút lưu thay đổi "Save" -> Người chơi nhấn vào một nút (ví dụ: leftKey) -> Hệ thống gọi StartAssignment("left") trong MenuScript.cs -> Trạng thái chờ phím được kích hoạt->Người chơi nhấn "Save" -> Hệ thống lưu tất cả thay đổi lại và chuyển về Scene Menu.
- Chức năng **điều khiển nhân vật**: Sau khi màn chơi được tải lên -> Người chơi sử dụng phím điều khiển tùy chỉnh để di chuyển nhân vật trái/phải hoặc nhảy.
 - o Khi nhấn phím di chuyển trái/phải -> Nhân vật di chuyển với tốc độ 7, nhân vật xoay hướng dựa trên chiều di chuyển.
 - o Khi nhấn phím nhảy-> Nhân vật nhảy với độ cao 10 và âm thanh nhảy được phát.
- Chức năng **tấn công kẻ địch**: Sau khi người chơi điều khiển nhân vật di chuyển trong màn chơi -> Gặp kẻ thù trên đường đi -> Người chơi nhấn phím tấn công. Khi nhấn phím tấn công -> Hệ thống chọn ngẫu nhiên một trong ba animation tấn công (Attack1, Attack2, Attack3)-> Nhân vật thực hiện đòn tấn công tại điểm tấn công trong phạm vi bán kính 1.5-> Nếu có kẻ thù trong phạm vi kẻ thù nhận 1 sát thương. Khi kẻ thù bị trúng đòn -> Hiện thị văn bản nổi tại vị trí kẻ thù để hiện số sát thương -> Kẻ thù bị trừ máu -> Nếu máu kẻ thù về 0, kẻ thù bị phá hủy.
- Chức năng **thu thập vật phẩm**:

- Khi người chơi di chuyển nhân vật trong màn chơi -> Va chạm với đồng xu-> Số đồng xu của người chơi tăng lên 1-> Đồng xu bị phá hủy ngay lập tức Số đồng xu được cập nhật trên giao diện để người chơi biết số lượng hiện tại.
- Chức năng **tương tác với bẫy**: Khi người chơi di chuyển nhân vật trong màn chơi -> Va chạm với bẫy -> Nhân vật chết ngay lập tức. Khi nhân vật chết -> Hiện thị màn hình game over , phát hiệu ứng nổ, rung màn hình , và phá hủy nhân vật.
- Chức năng **tương tác với chìa khóa**: Khi người chơi va chạm với chìa khóa -> Hệ thống phát hiện va chạm qua-> Màn hình chiến thắng hiện lên, trạng thái chiến thắng được kích hoạt, chìa khóa bị phá hủy, và nhân vật dừng di chuyển.
- Chức năng **nhận sát thương và game over**: Khi nhân vật va chạm với kẻ thù hoặc bẫy -> Hệ thống trừ máu-> Máu được cập nhật trên giao diện. Hiệu ứng rung màn hình được kích hoạt để tạo cảm giác bị tấn công. Nếu máu của nhân vật về 0-> Nhân vật chết, hiển thị màn hình game over, phát hiệu ứng nổ, rung màn hình mạnh hơn, và nhân vật bị phá hủy.

3.1.2 Mô tả hệ thống bằng UML

- Các từ khóa được dùng trong trò chơi

Bảng 3: Bảng từ khóa

STT	Tên	Mô tả
1	Chiến binh	Nhân vật chính mà người chơi điều khiển trong game. Là một chiến binh dũng cảm, có nhiệm vụ vượt qua các thử thách, chiến đấu với kẻ thù, thu thập đồng xu, và tìm chìa khóa để phá bỏ lời nguyền cứu vương quốc. Chiến binh sở hữu các kỹ năng cơ bản như di chuyển, nhảy, và tấn công.
2	Vương quốc	Bối cảnh chính của trò chơi, một vương quốc cổ xưa bị nguyền rủa. Đây là nơi diễn ra các màn chơi, với các khu rừng bí ẩn, nền tảng, và các công trình liên quan đến cốt truyện.
3	Lời nguyền	Cốt truyện trung tâm, đại diện cho thử thách mà chiến binh phải đối mặt. Lời nguyền được phá bỏ khi nhân vật tìm thấy chìa khóa vàng và đánh bại thủ lĩnh undead.

4	Kẻ địch	Các đối thủ mà chiến binh phải chiến đấu, bao gồm quái vật rừng, tiểu quái, và thủ lĩnh undead. Chúng xuất hiện trong màn chơi để cản trở người chơi và gây sát thương khi va chạm.
5	Quái nắm	Loại kẻ thù cơ bản, di chuyển qua lại trên nền tảng và tấn công khi nhân vật tiếp cận. Đại diện cho các mối nguy hiểm đầu tiên trong hành trình của chiến binh.
6	Tiểu quái/ minion	Kẻ thù nhỏ xuất hiện thành nhóm, di chuyển nhanh và tấn công bằng cách lao thẳng. Chúng dễ bị tiêu diệt nhưng xuất hiện với số lượng lớn, tăng độ khó.
7	Boss	Trùm cuối của màn chơi, bảo vệ chìa khóa vàng. Có sức mạnh lớn, khả năng triệu hồi tiểu quái và tấn công bằng sóng năng lượng, là thử thách lớn nhất của người chơi.
8	Coin	Vật phẩm có thể thu thập trong game, đại diện cho điểm số hoặc tài nguyên. Mỗi khi thu thập, số lượng đồng xu tăng lên và được hiển thị trên giao diện.
9	Key	Vật phẩm quan trọng để hoàn thành màn chơi. Khi chiến binh lấy được chìa khóa, trò chơi kết thúc với màn hình chiến thắng, phá bỏ được 1 lời nguyền.
10	Bẫy	Các chướng ngại vật nguy hiểm trong màn chơi, gây sát thương hoặc khiến nhân vật chết ngay lập tức khi va chạm. Đại diện cho các mối nguy ngoài kẻ thù.
11	Mặt đất	Nơi mà chiến binh di chuyển và nhảy lên. Là yếu tố quan trọng để xác định trạng thái nhảy của nhân vật.
12	Di chuyển	Hành động cơ bản của chiến binh, bao gồm nhảy, đi trái, đi phải, và xoay hướng dựa trên điều khiển của người chơi.
13	Tấn công	Hành động chiến đấu của chiến binh, sử dụng các đòn tấn công ngẫu nhiên để gây sát thương cho kẻ thù trong phạm vi nhất định.
14	Sát thương	Lượng máu bị giảm khi chiến binh hoặc kẻ thù bị tấn công. Được hiển thị qua văn bản nổi và ảnh hưởng đến trạng thái sống/chết.

15	Máu	Chỉ số sức khỏe của chiến binh và kẻ thù. Khi máu về 0, nhân vật hoặc kẻ thù bị phá hủy.
16	Chiến thắng	Trạng thái khi chiến binh lấy được chìa khóa, dẫn đến màn hình chiến thắng và kết thúc thành công màn chơi.
17	Thua cuộc	Trạng thái khi chiến binh chết (máu về 0 hoặc va chạm với bẫy), dẫn đến màn hình thua cuộc.
18	Menu	Giao diện chính cho người chơi, bao gồm các tùy chọn như gán phím và thoát game.
19	Keyboard control	Tính năng cho phép người chơi tùy chỉnh phím điều khiển (trái, phải, nhảy, tấn công) thông qua Scene riêng. Có thể bật nhanh thông qua nút esc.
20	Lưu trữ	Quá trình lưu các phím điều khiển đã gán để sử dụng trong các lần chơi sau, sử dụng hệ thống lưu trữ của game.
21	Hiệu ứng	Các hiệu ứng trực quan và âm thanh trong game, như nổ khi chết, văn bản nổi khi gây sát thương, rung màn hình khi bị tấn công, và âm thanh nhảy.
22	Rung màn hình	Hiệu ứng tạo cảm giác chân thực khi nhân vật nhận sát thương hoặc chết, tăng tính tương tác.
23	Âm thanh	Hiệu ứng âm thanh, như âm thanh nhảy, được phát khi nhân vật thực hiện hành động nhảy.
24	Giao diện	Các yếu tố hiển thị trên màn hình, bao gồm số đồng xu, máu, và các thông báo (chiến thắng, thua cuộc).
25	Màn chơi	Khu vực chính nơi diễn ra gameplay, bao gồm Level1 màn khác
26	maxHealth	Lượng máu tối đa của nhân vật.
27	currentCoin	Lượng xu thu thập hiện tại của nhân vật.

3.1.3 Kịch bản chuẩn và ngoại lệ

a, Scenario chức năng Thu thập item khi quái chết

Bảng 4 : Bảng Scenario Thu thập item khi quái chết

Scenario	Thu thập item khi quái chết
----------	-----------------------------

Actor	Người chơi
Tiền điều kiện	Người chơi đã tham gia vào game, kẻ địch(quái) đã chết.
Mục đích	Cung cấp phần thưởng cho người chơi sau khi tiêu diệt kẻ địch.
Main Event	<ol style="list-style-type: none"> 1. Kẻ địch đã bị người chơi tiêu diệt. 2. Sinh item. <ul style="list-style-type: none"> – Chọn ngẫu nhiên giữa coin và heart (xác suất 50/50). – item xuất hiện tại vị trí hiện tại của kẻ địch. – Đặt thời gian sống cho item (5 giây) trước khi tự phá hủy. 3. Player thu thập item: <ul style="list-style-type: none"> – Nếu item là coin: Tăng currentCoin của Player lên 1. Phá hủy item. – Nếu item là heart: Nếu máu hiện tại của nhân vật < maxHealth, tăng máu lên 1 (không vượt quá maxHealth). 4. Phá hủy item. 5. Cập nhật UI để hiển thị currentCoin và currentHealth.
Ngoại lệ	<p>2.1 Item tồn tại quá lâu: Player không thu thập item trong 5 giây. Item tự phá hủy, không ảnh hưởng game.</p> <p>3.1 Player không thu thập được item heart khi: máu hiện tại vượt quá maxHealth.</p>

b, Scenario chức năng tấn công kẻ địch

Bảng 5 : Bảng Scenario chức năng tấn công kẻ địch

Scenario	Tấn công kẻ địch
Actor	Người chơi
Tiền điều kiện	Người chơi đã tham gia vào game
Mục đích	Người chơi tấn công kẻ địch trong phạm vi nhất định để gây sát thương.
Main Event	<ol style="list-style-type: none"> 1. Sau khi đã vào trong trận đấu, người chơi đã nhấn nút phím tấn công hoặc hệ thống đã cập nhật các nút từ lần chơi gần đây nhất. 2. Hệ thống kiểm tra thời gian chờ để đảm bảo không tấn công quá nhanh.

	<p>3. Player gọi animator.SetTrigger("Attack") để phát hoạt hình tấn công.</p> <p>4. Kiểm tra va chạm: nếu phát hiện kẻ địch trong phạm vi tấn công.</p> <ul style="list-style-type: none"> – Người chơi nhận được thông tin kẻ thù đã bị sát thương. – Hiệu ứng sát thương – Rung camera – Hiện thị văn bản nổi <p>5. Kẻ địch phản ứng: giảm máu, chết nếu máu về 0.</p>
Ngoại lệ	<p>2.1 Thời gian hồi chiêu chưa hết, yêu cầu tấn công không được thực hiện</p> <p>4.1 Không có kẻ địch trong phạm vi:</p> <ul style="list-style-type: none"> – Không thực hiện hành động gây sát thương. – Không phát hiệu ứng rung camera hoặc floating text.

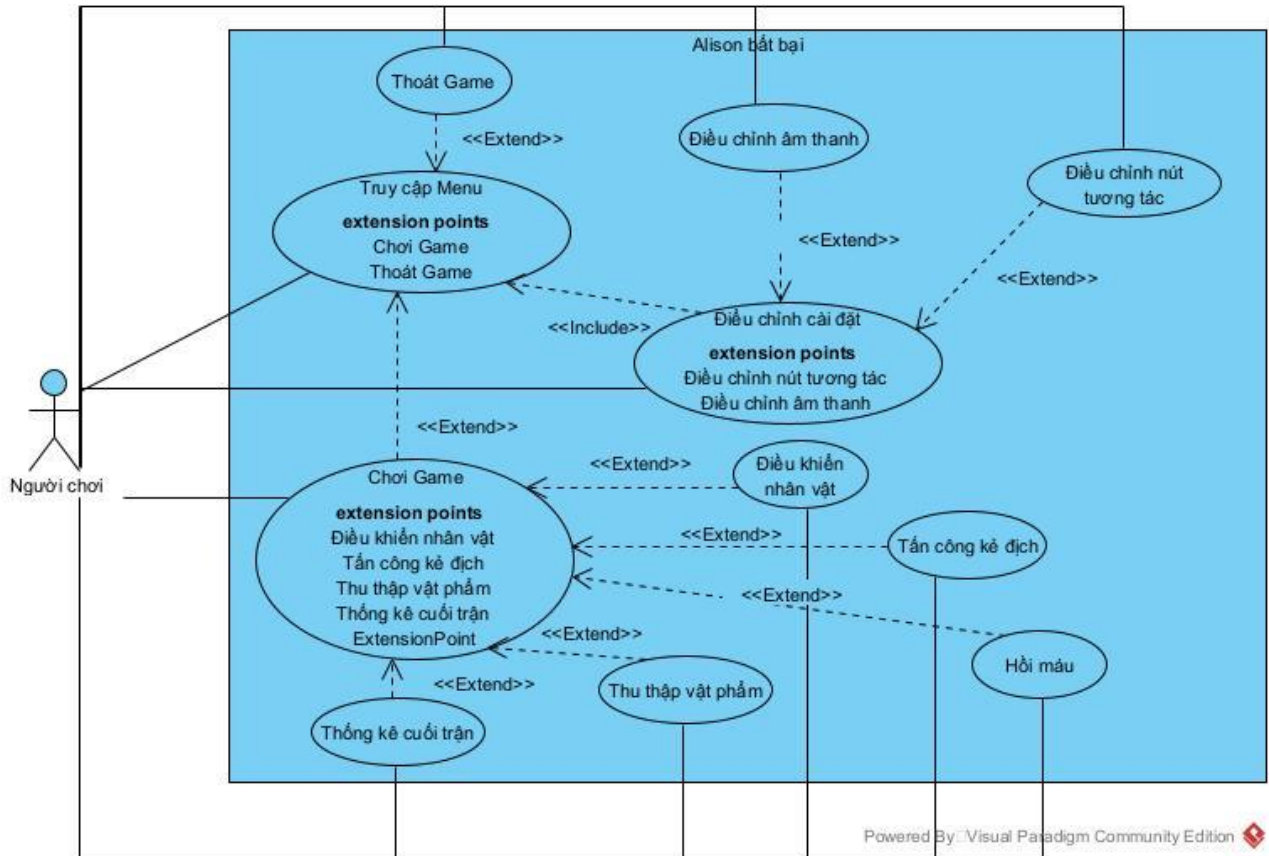
c, Scenario chức năng Gán phím tương tác cho nhân vật

Bảng 6 : Bảng Scenario chức năng Gán phím tương tác cho nhân vật

Scenario	Gán phím
Actor	Người chơi
Tiền điều kiện	Người chơi đã tham gia vào trận đấu, truy cập scene “Set” thông qua menu chính
Mục đích	Gán phím di chuyển, tấn công theo ý thích người chơi.
Main Event	<p>1. Sau khi đã vào trong màn hình chính, chọn “Set”.</p> <p>2. Màn hình cài đặt game hiện ra với các phím gán mặc định sẵn.</p> <ul style="list-style-type: none"> – Di chuyển phải : D – Di chuyển trái : A – Nhảy : space – Tấn công : Q. <p>3. Người dùng nhấp vào phím tương tác muốn thay đổi và nhập vào phím bản thân muốn.</p> <p>4. Thay đổi xong tiếp tục nhấn vào Set để trở về màn hình chính</p> <p>5. Hệ thống ghi nhận cập nhật thay đổi.</p>
Ngoại lệ	3.1 Nhập phím không hợp lệ ví dụ dùng nháy chuột trái/ phải: hệ thống không ghi nhận, gán phím vẫn giữ nguyên.

3.2 Người dùng nhập chuỗi chữ cái: hệ thống ghi nhận chữ cái đầu tiên được nhập.

3.1.4. Usecase tổng quan



Hình 1: Sơ đồ usecase tổng quan

Mô tả Usecase tổng quan:

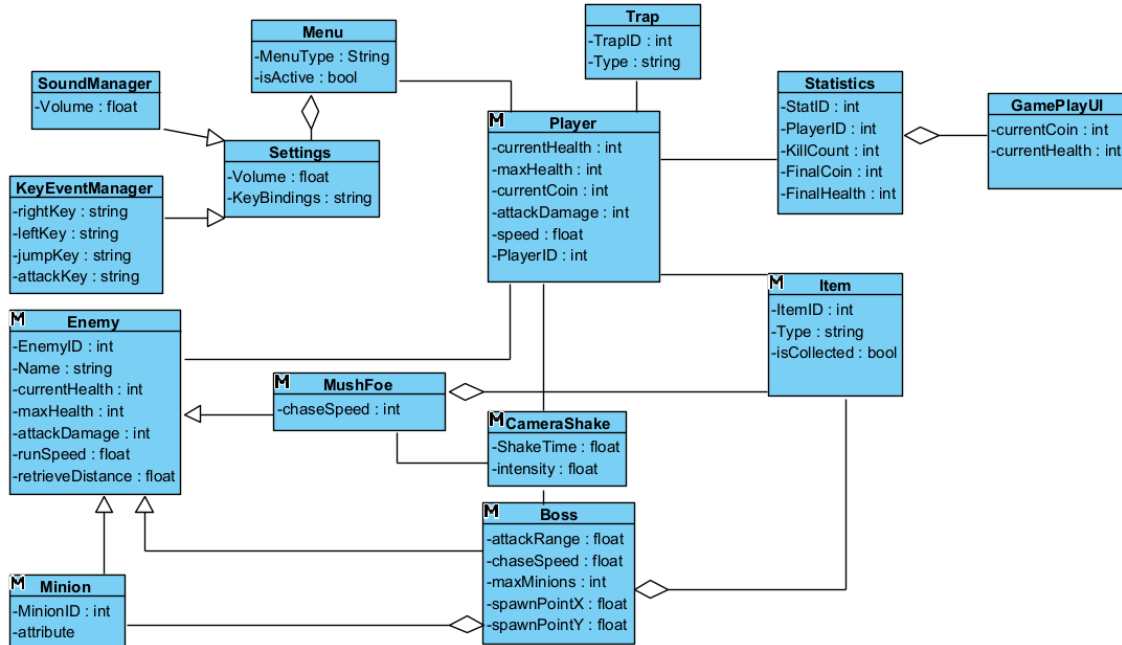
- UC Thoát Game: Cho phép người chơi thoát game.
- UC Truy cập Menu: Cho phép người dùng truy cập để điều chỉnh các tác vụ của game hoặc thoát thông qua màn hình Menu.
- UC Điều chỉnh cài đặt: yêu cầu người chơi phải truy cập thông qua màn hình truy cập của Menu. Bao gồm điều chỉnh âm thanh và nút tương tác, giúp cá nhân hóa trải nghiệm. Thay đổi được lưu trước khi thoát.:
 - o UC Điều chỉnh âm thanh: Cho phép người chơi điều chỉnh âm lượng game. Sử dụng Slider trong menu để thay đổi. Tăng cường trải nghiệm âm thanh theo sở thích.

- UC Điều chỉnh nút tương tác: Cho phép người chơi tùy chỉnh phím chức năng. Người chơi gán phím mới trong menu, lưu cấu hình để phù hợp với phong cách chơi. Giúp tối ưu hóa thao tác.
- UC Chơi Game: Giúp người chơi tham gia hành trình game trực tiếp. Bao gồm:
 - UC Điều khiển nhân vật: Cho phép người chơi di chuyển nhân vật trên bản đồ. Sử dụng phím đã được gán để điều hướng, giúp khám phá và chiến đấu hiệu quả.
 - UC Tấn công kẻ địch: Cho phép người chơi tấn công kẻ địch để tiêu diệt. Nhấn nút đã gán để gây sát thương, kèm hiệu ứng rung camera và floating text. Hỗ trợ tiến trình chiến đấu thêm sinh động.
 - UC Thu thập vật phẩm: Cho phép người chơi thu thập coin và heart từ kẻ địch chết hoặc trong quá trình di chuyển, giúp để tăng tài nguyên hoặc hồi máu, tối ưu hóa sức mạnh. Tăng tính chiến lược.
 - UC Hồi máu: Cho phép người chơi thu thập máu (heart) trong chiến đấu để hồi phục sức khỏe. Hỗ trợ kéo dài thời gian sống sót trong trận đấu.
 - UC Thống kê cuối trận: Cho phép người chơi xem thông tin sau trận đấu. Hiện thị coin, mạng giết, lượng máu còn lại, giúp đánh giá hiệu quả. Hỗ trợ cải thiện chiến thuật cho lần sau.

3.2 Thiết kế hệ thống trò chơi

3.2.1 Sơ đồ lớp thực thể hệ thống

Hình 2: Biểu đồ lớp thực thể tổng quan của trò chơi



Bảng 7: Bảng mô tả tổng quan các lớp liên quan

Stt	Lớp	Mô tả
1	SoundManager	Quản lý âm thanh trong game, bao gồm điều chỉnh âm lượng và lưu cài đặt.
2	Menu	Quản lý các giao diện menu
3	Settings	Lưu trữ và quản lý các cài đặt game (âm thanh, nút tương tác).
4	Player	Đại diện cho người chơi, quản lý trạng thái và hành động của nhân vật chính.
5	Statistics	Quản lý và hiển thị thông tin thống kê cuối trận.
6	Item	Đại diện cho các vật phẩm (Coin, Heart) sinh ra từ kẻ địch chết hoặc quá trình khám phá của player.
7	CameraShake	Quản lý hiệu ứng rung camera khi tấn công, nhân vật hoặc kẻ địch chết.

8	GamePlayUI	Quản lý giao diện người dùng trong gameplay (UI hiển thị máu, coin, thống kê).
9	Enemy	Lớp cha (abstract) đại diện cho tất cả kẻ địch (bao gồm Minion, Boss, MushFoe)
10	Minion	Lớp con của Enemy, đại diện cho kẻ địch yếu, thường được Boss triệu hồi.
11	Boss	Lớp con của Enemy, đại diện cho kẻ địch mạnh (Boss).
12	MushFoe	Lớp con của Enemy, quản lý quái nấm chuyên đi tuần tra ở khu vực nhất định.
13	KeyEventManager	Quản lý các sự kiện gõ phím trong game.
14	Trap	Đại diện cho các bẫy trong game.

Bảng 8: Bảng mô tả lớp SoundManager

Stt	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	Volume	float	Đại diện cho mức âm lượng của game, được điều chỉnh thông qua Slider trong Menu.

Bảng 9: Bảng mô tả lớp Menu

Stt	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	MenuType	string	Phân loại các giao diện menu để xử lý hành động phù hợp (Menu vào game, menu kết thúc game, menu chỉnh sửa).
2	isActive	Bool	Điều khiển việc bật/tắt menu trên giao diện người dùng.

Bảng 10: Bảng mô tả lớp Settings

Stt	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	Volume	float	Giá trị âm lượng được lưu trữ từ Sound Manager. Lưu trữ cài đặt âm thanh để đồng bộ hóa giữa các phiên chơi.
2	KeyBindings	string	Chuỗi chứa cấu hình phím (ví dụ: left, right, jump, attack). Lưu các phím đã gán để Player sử dụng trong điều khiển nhân vật.

Bảng 11: Bảng mô tả lớp Player

Stt	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	currentHealth	int	Sức khỏe hiện tại của người chơi. Theo dõi trạng thái sống/chết và hiển thị trên UI.
2	maxHealth	int	Sức khỏe tối đa của người chơi. Giới hạn tối đa khi hồi máu.
3	currentCoin	int	Số coin hiện tại mà người chơi sở hữu.
4	attackDamage	int	Sát thương cơ bản mà người chơi gây ra khi tấn công. Quy định sức mạnh tấn công của Player.
5	speed	float	Tốc độ di chuyển của nhân vật.

Bảng 12: Bảng mô tả lớp Statistics

Stt	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	StatID	int	ID duy nhất của bản thống kê. Định danh từng bản thống kê cho từng trận đấu.
3	KillCount	int	Số mạng kẻ địch mà người chơi đã tiêu diệt. Theo dõi hiệu suất chiến đấu.
4	FinalCoin	int	Đánh giá tài nguyên cuối trận.

5	FinalHealth	int	Sức khỏe còn lại khi trận đấu kết thúc. Đánh giá trạng thái sống sót.
---	-------------	-----	--

Bảng 13: Bảng mô tả lớp GameplayUI

Stt	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	currentCoin	int	Số coin hiện tại của người chơi, được đồng bộ từ Player. Hiển thị trên UI để người chơi theo dõi.
2	currentHealth	int	Sức khỏe hiện tại của người chơi, được đồng bộ từ Player. Hiển thị trên UI để theo dõi trạng thái.

Bảng 14: Bảng mô tả lớp Enemy

Stt	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	EnemyID	int	ID duy nhất của kẻ địch. Định danh từng kẻ địch trong hệ thống.
2	Name	string	Tên của kẻ địch (ví dụ: Enemy1, Undead_Boss, minions). Phân biệt loại kẻ địch.
3	currentHealth	int	Sức khỏe hiện tại của kẻ địch. Theo dõi trạng thái sống/chết.
4	maxHealth	int	Sức khỏe tối đa của kẻ địch. Đặt giới hạn sức khỏe ban đầu.
5	attackDamage	int	Sát thương mà kẻ địch gây ra khi tấn công. Quy định sức mạnh tấn công của kẻ địch.
6	runSpeed	float	Tốc độ di chuyển cơ bản của kẻ địch.

Bảng 15: Bảng mô tả lớp Minion

Stt	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	MinionID	int	ID duy nhất của Minion. Định danh từng Minion được triệu hồi.

Bảng 16: Bảng mô tả lớp Boss

Stt	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	attackRange	float	Phạm vi tấn công của Boss.
2	chaseSpeed	float	Tốc độ đuổi theo Player khi trong tầm nhìn.
3	maxMinions	int	Số lượng Minion tối đa mà Boss có thể triệu hồi. Giới hạn số Minion trên sân.
4	spawnPointX	float	Tọa độ X của điểm triệu hồi Minion. Xác định vị trí sinh Minion.
5	spawnPointY	float	Tọa độ Y của điểm triệu hồi Minion. Xác định vị trí sinh Minion.

Bảng 17: Bảng mô tả lớp MushFoe

Stt	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	chaseSpeed	float	Tốc độ đuổi theo Player của MushFoe.

Bảng 18: Bảng mô tả lớp Item

Stt	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	ItemID	int	ID duy nhất của vật phẩm. Định danh từng vật phẩm (Coin, Heart).
2	Type	string	Loại vật phẩm (ví dụ: Coin, Heart).

3	isCollected	bool	Trạng thái đã được thu thập (true nếu đã lấy, false nếu chưa). Quản lý việc vật phẩm có thể bị thu thập nhiều lần.
---	-------------	------	--

Bảng 19: Bảng mô tả lớp CameraShake

Stt	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	ShakeTime	float	Thời gian rung của camera. Quy định độ dài hiệu ứng rung khi tấn công hoặc nhân vật/quái chết.
2	intensity	float	Độ mạnh của hiệu ứng rung. Điều chỉnh mức độ rung để tăng tính kịch tính.

Bảng 20: Bảng mô tả lớp KeyEventManager

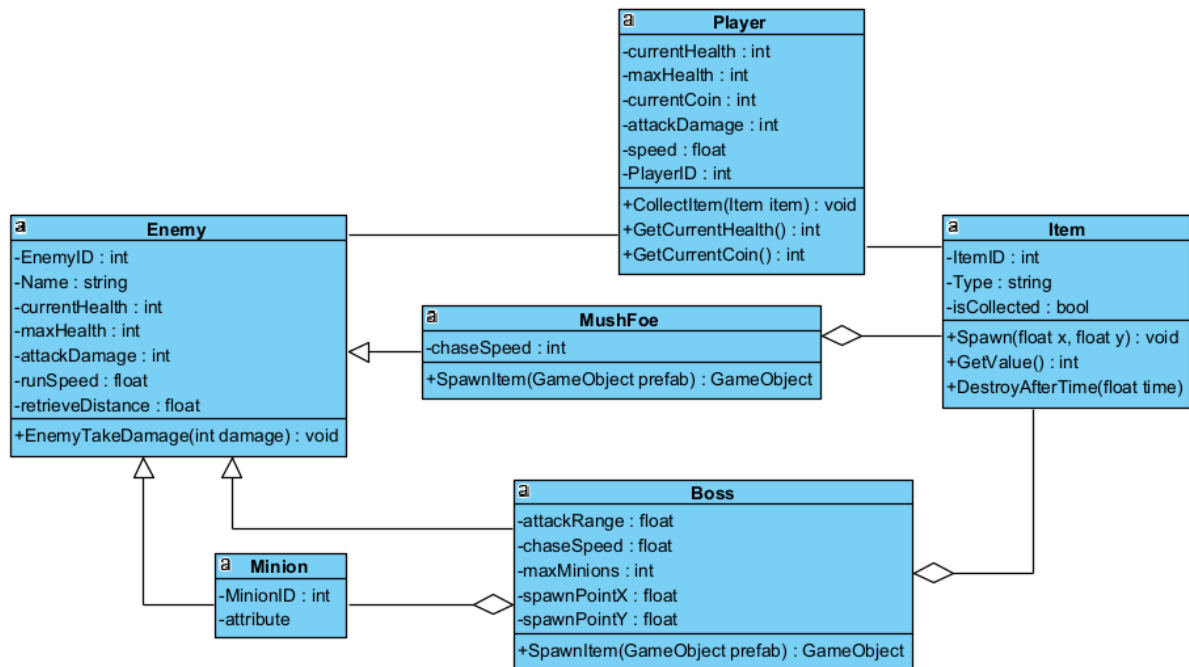
Stt	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	rightKey	string	Gán phím di chuyển phải.
2	leftKey	string	Gán phím di chuyển trái.
3	jumpKey	string	Gán phím nhảy.
4	attackKey	string	Gán phím tấn công.

Bảng 21: Bảng mô tả lớp Trap

Stt	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	Type	string	Phân loại các giao diện menu để xử lý hành động phù hợp (Menu vào game, menu kết thúc game, menu chỉnh sửa).
2	TrapID	int	ID duy nhất của bẫy. Định danh từng bẫy trên bản đồ.

3.2.2 Sơ đồ lớp module Thu thập item khi quái chết

Hình 3: Biểu đồ lớp module Thu thập item khi quái chết



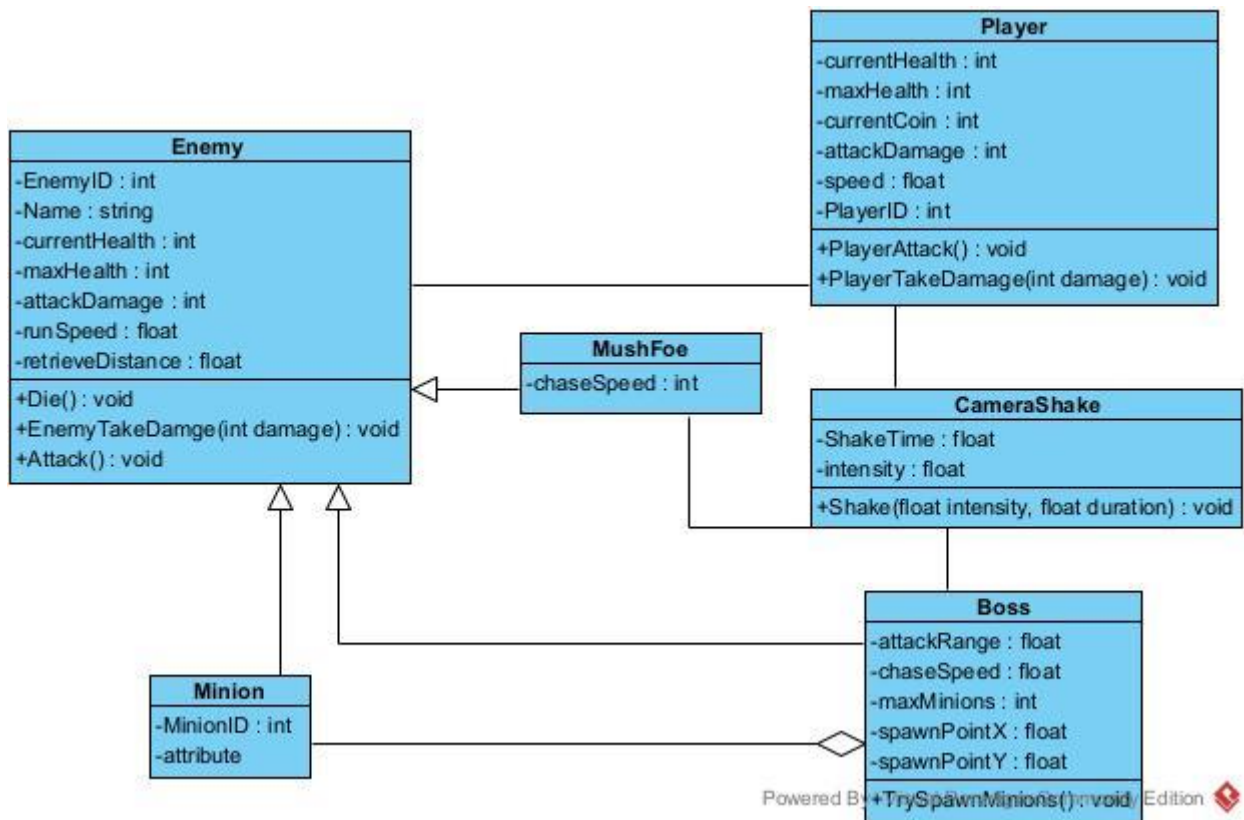
Bảng 22: Bảng mô tả các phương thức của lớp module Thu thập item khi quái chết

Stt	Tên phương thức	Kiểu trả về	Mô tả
1	CollectItem(Item item)	void	Thu thập vật phẩm khi Player va chạm với Item.
2	EnemyTakeDamage(int damage)	void	Hàm gây sát thương lên kẻ địch nói chung.
3	GetCurrentHealth()	int	Cập nhật giá trị sức khỏe hiện tại của người chơi.
	GetCurrentCoin()	int	Cập nhật giá trị coin hiện tại của người chơi.
	SpawnItem(GameObject prefab)	GameObject	Sinh một vật phẩm tại vị trí khi kẻ địch chết
	Spawn(float x, float y)	void	Sinh vật phẩm tại vị trí cho trước.

	GetValue()	int	Trả về giá trị của vật phẩm (số coin hoặc lượng máu hồi).
	DestroyAfterTime(float time)	void	Phá hủy vật phẩm sau một khoảng thời gian nếu không được thu thập.

3.2.3 Sơ đồ lớp module Tấn công kẻ địch thiết kế.

Hình 4: Biểu đồ lớp module Tấn công kẻ địch



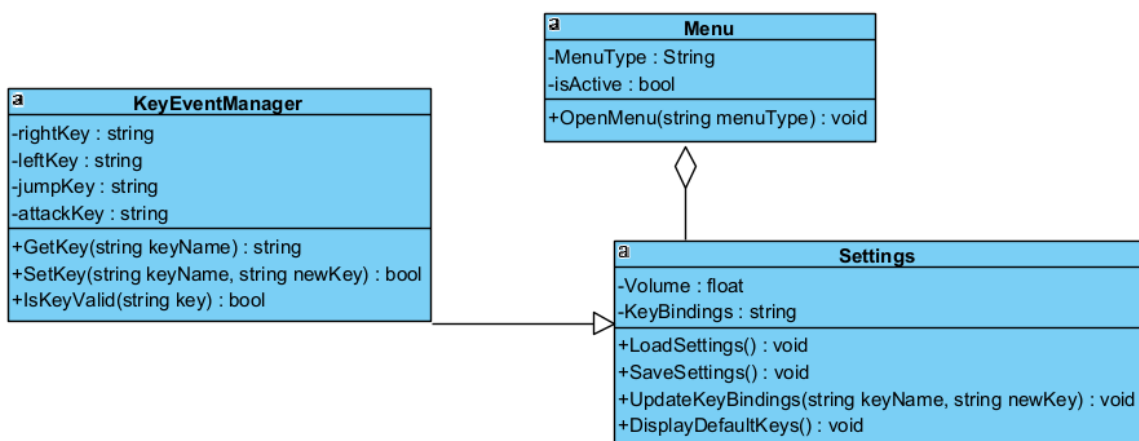
Bảng 23: Bảng mô tả các phương thức của lớp module Tấn công địch

Stt	Tên phương thức	Kiểu trả về	Mô tả
1	Die()	void	Xử lý hành vi khi kẻ địch chết, bao gồm phá hủy GameObject và sinh vật phẩm.
2	EnemyTakeDamage(int damage)	void	Hàm gây sát thương lên kẻ địch nói chung.
3	PlayerAttack()	Void	Thực hiện hành động tấn công của người chơi, nhắm vào các kẻ địch trong phạm vi

			attackRadius. Gọi hàm gây sát thương trên các đối tượng trong phạm vi, và kích hoạt hiệu ứng (như CameraShake).
4	PlayerTakeDamage(int damage)	Void	Hàm gây sát thương lên người chơi
5	TrySpawnMinions()	Void	Thử triệu hồi Minion nếu số lượng hiện tại chưa đạt maxMinions.
6	Shake(float intensity, float duration)	void	Thực hiện hiệu ứng rung camera với độ mạnh và thời gian đã cho.

3.2.3 Sơ đồ lớp module Gán phím tương tác cho nhân vật.

Hình 5: Biểu đồ lớp module Gán phím tương tác cho nhân vật



Bảng 24: Bảng mô tả các phương thức trong lớp module Gán phím

Stt	Tên phương thức	Kiểu trả về	Mô tả
1	GetKey(string keyName)	string	Trả về giá trị phím (string) tương ứng với tên phím được yêu cầu (leftKey, rightKey, jumpKey, attackKey).

2	SetKey(string keyName, string newKey)	bool	Cập nhật giá trị phím mới cho tên phím được chỉ định, trả về true nếu thành công, false nếu thất bại (trùng lặp hoặc không hợp lệ).
3	IsValidKey(string key)	bool	Kiểm tra xem phím nhập vào có hợp lệ hay không (chỉ chấp nhận phím đơn, không phải tổ hợp).
4	OpenMenu(string menuType)	Void	Mở menu tương ứng dựa trên loại menu được chỉ định (ví dụ: "Settings"), đặt isActive = true.
5	LoadSettings()	Void	Tải cài đặt phím từ PlayerPrefs
6	SaveSettings()	void	Lưu các phím hiện tại trong keyBindings vào PlayerPrefs sau khi thay đổi.
7	UpdateKeyBindings(string keyName, string newKey)	void	Cập nhật keyBindings với phím mới cho tên phím được chỉ định và thông báo KeyEventManager.
8	DisplayDefaultKeys()	void	Hiển thị các phím mặc định hoặc hiện tại đang được gán trên giao diện.

CHƯƠNG 4: CÁC MÀN HÌNH GIAO DIỆN TRONG GAME

Hình 6: Màn hình Menu chính khi vào game



Hình 7: Màn hình cài đặt của game



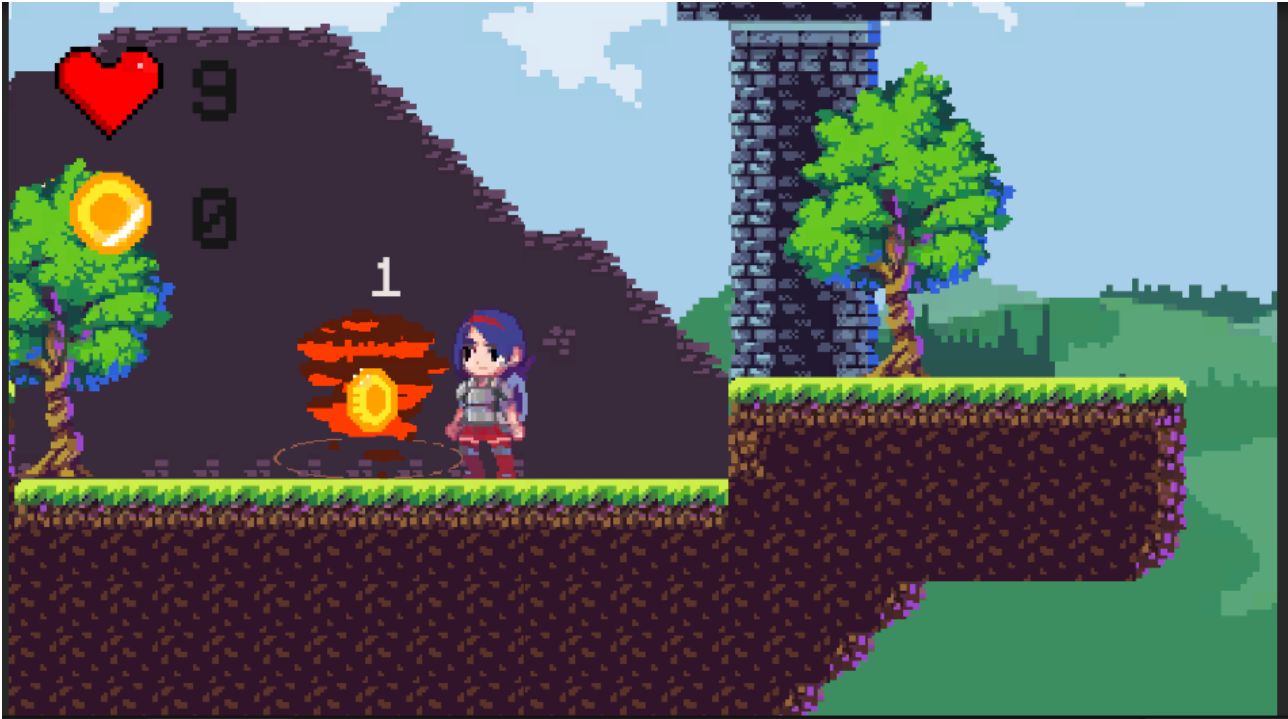
Hình 8: Màn hình giao diện khi bắt đầu vào chơi game



Hình 9 : Màn hình khi chiến đấu kẻ địch (quái nấm)

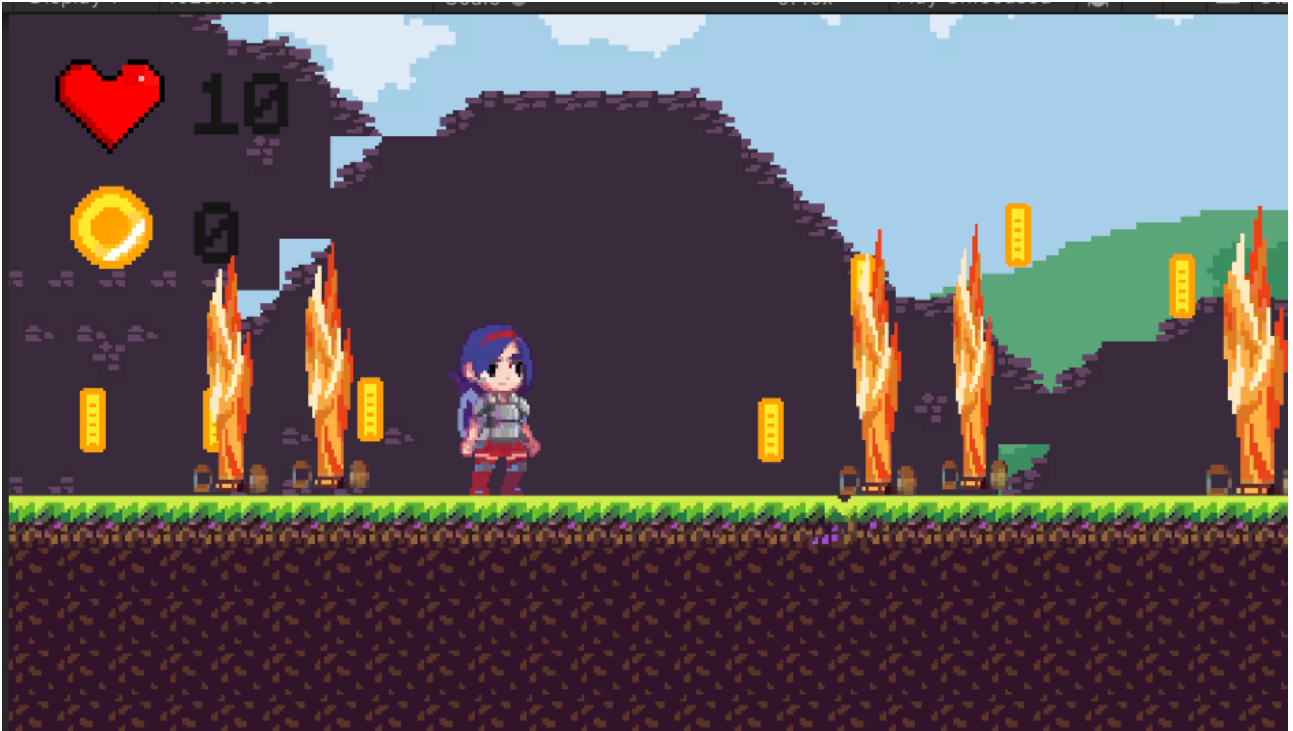


Hình 10: Màn hình khi quái bị hạ xuất hiện hiệu ứng nổ cùng vật phẩm thưởng



Hình 11: Màn hình các loại bẫy xuất hiện trong game

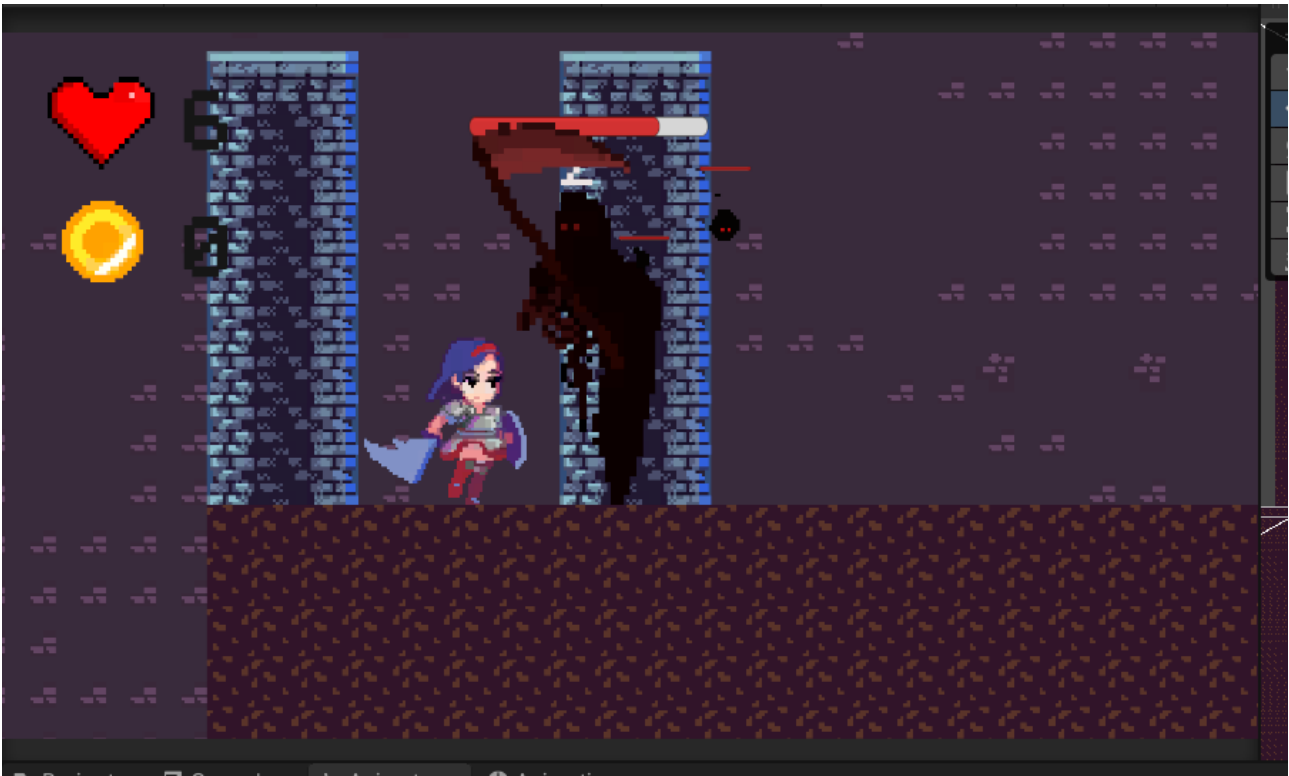
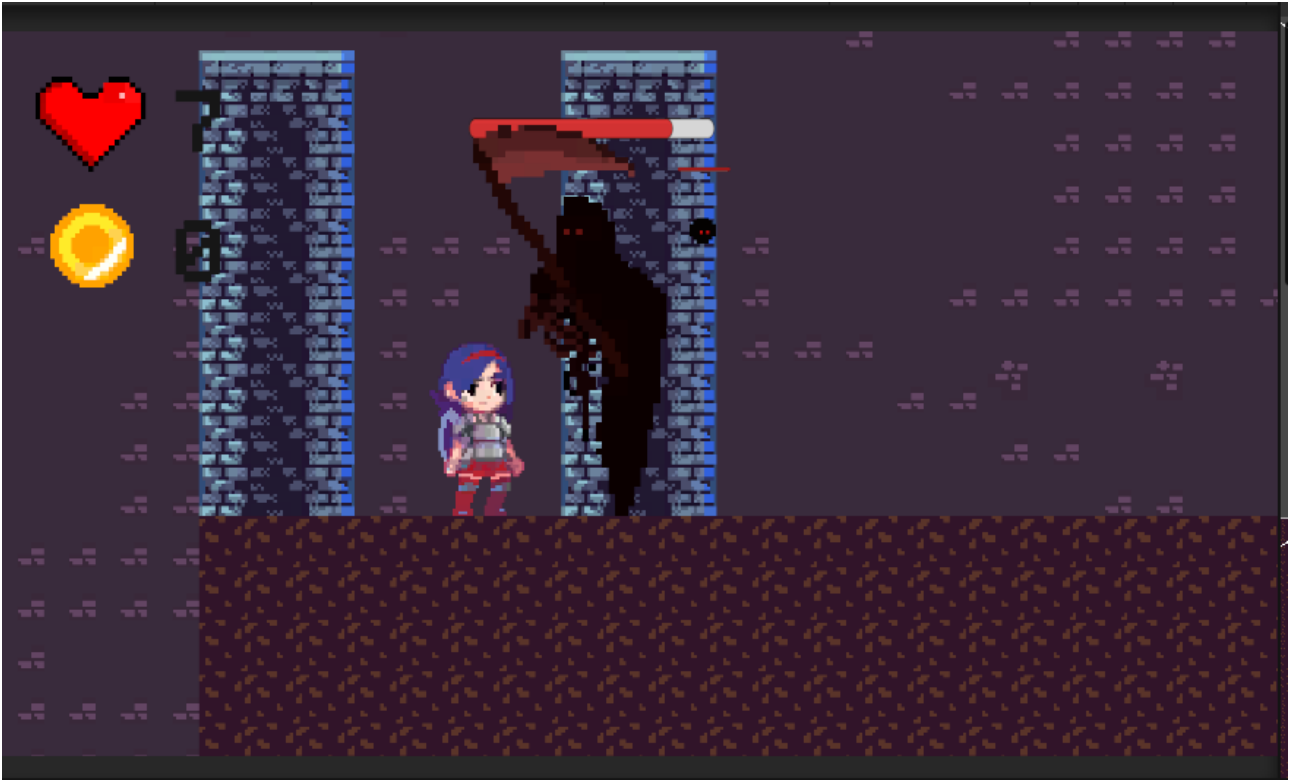




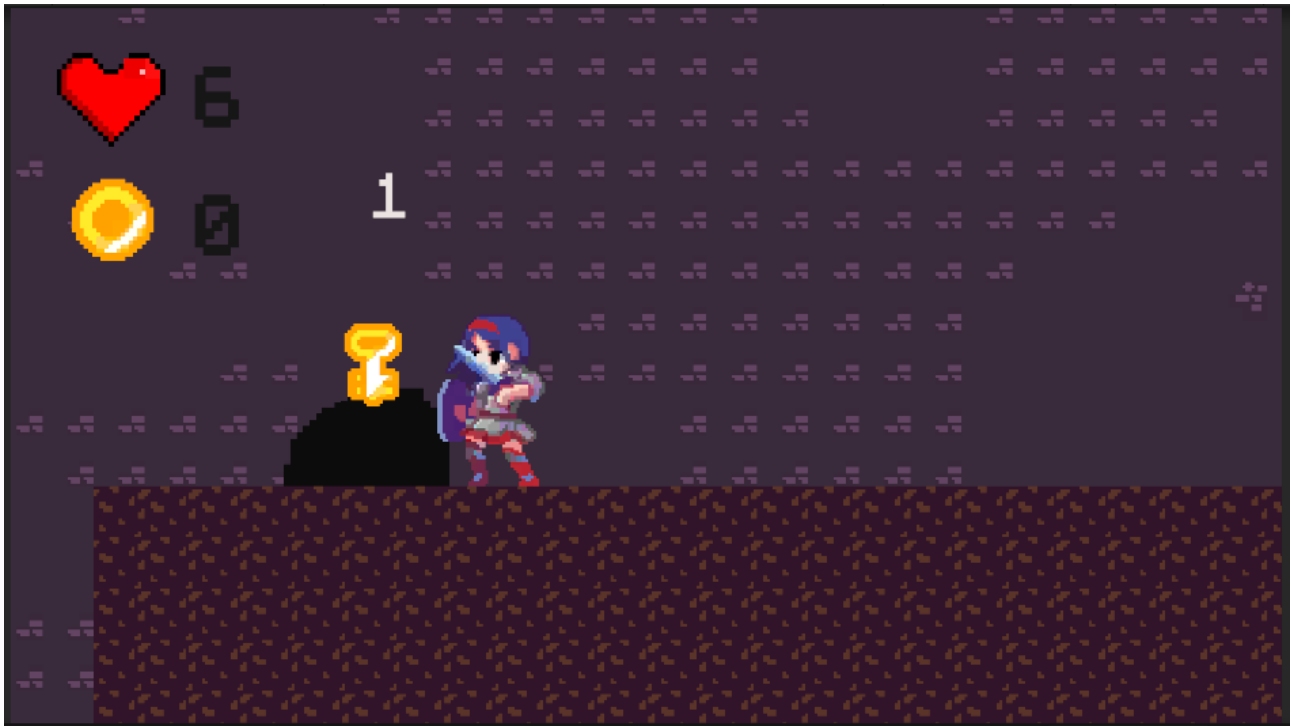
Hình 12: Màn hình khi gặp Boss game



Hình 13: Màn hình boss triệu hồi minion tấn công



Hình 14: Màn hình nhận được chìa khóa sau khi boss bị hạ



Hình 15: Màn hình khi thua game



Hình 16: Màn hình hiển thị thắng khi thu được chìa khóa vàng

