

BỘ GIÁO DỤC VÀ ĐÀO TẠO

BỘ NÔNG NGHIỆP VÀ PTNT

TRƯỜNG ĐẠI HỌC THỦY LỢI

-----□□□□-----



HỌ VÀ TÊN: PHẠM THANH SƠN

PHÁT TRIỂN GAME THE KNIGHT RETURN

ĐỒ ÁN TỐT NGHIỆP

Hà Nội, Năm 2024

BỘ GIÁO DỤC VÀ ĐÀO TẠO BỘ NÔNG NGHIỆP VÀ PTNT
TRƯỜNG ĐẠI HỌC THỦY LỢI



PHẠM THANH SƠN

PHÁT TRIỂN GAME THE KNIGHT RETURN

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, NĂM 2024

BỘ GIÁO DỤC VÀ ĐÀO TẠO BỘ NÔNG NGHIỆP VÀ PTNT
TRƯỜNG ĐẠI HỌC THỦY LỢI



PHẠM THANH SƠN

PHÁT TRIỂN GAME THE KNIGHT RETURN

Ngành: Công nghệ thông tin

Mã số: 7480201

NGƯỜI HƯỚNG DẪN: PGS.TS LÊ ĐỨC HẬU

HÀ NỘI, NĂM 2024



CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Phạm Thanh Sơn

Hệ đào tạo: Đại học chính quy

Lớp: 62TH-NB

Ngành: Công nghệ thông tin

Khoa: Công nghệ thông tin

1. TÊN ĐỀ TÀI: Phát triển game The Knight Return

2. CÁC TÀI LIỆU CƠ BẢN:

[1] Unity Technologies, "Unity User Manual," Unity, 2023. [Online]. Available: <https://docs.unity3d.com/Manual/index.html>. [Accessed 05 07 2024].

[2] J. Hocking, "Unity in Action: Multiplatform Game Development in C# with Unity 5," Manning Publications ,2015 . [Online]. Available: <https://www.manning.com/books/unity-in-action>. [Accessed 05 07 2024].

[3] M. Geig, "Unity Game Development in 24 Hours, Sams Teach Yourself," Sams Publishing, 2013. [Online]. Available: <https://www.pearson.com/store/p/unity-game-development-in-24-hours-sams-teach-yourself/P100000022309>. [Accessed 05 07 2024].

3. NỘI DUNG CÁC PHẦN THUYẾT MINH VÀ TÍNH TOÁN:

Nội dung cần thuyết minh	Tỉ lệ %
Chương 1: Cơ sở lý thuyết và công nghệ phát triển game	25%
Chương 2: Phân tích và thiết kế hệ thống	25%
Chương 3 : Tổng quan về game The Knight Return	40%
Chương 3 : Cài đặt trò chơi và kết luận	10%

4. GIÁO VIÊN HƯỚNG DẪN TƯNG PHẦN

Phần	Họ và tên giáo viên hướng dẫn
Chương 1: Cơ sở lý thuyết và công nghệ phát triển game	PGS.TS. Lê Đức Hậu
Chương 2: Phân tích và thiết kế hệ thống	PGS.TS. Lê Đức Hậu
Chương 3 : Tổng quan về game The Knight Return	PGS.TS. Lê Đức Hậu
Chương 4: Cài đặt trò chơi và kết luận	PGS.TS. Lê Đức Hậu

5. NGÀY GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Ngày.....tháng.....năm 202...

Trưởng Bộ môn (Ký và ghi rõ Họ tên)	Giáo viên hướng dẫn chính (Ký và ghi rõ Họ tên)
--	--

Nhiệm vụ Đồ án tốt nghiệp đã được Hội đồng thi tốt nghiệp của Khoa thông qua.

Ngày.....tháng.....năm 202...

Chủ tịch Hội đồng
(Ký và ghi rõ Họ tên)

Sinh viên đã hoàn thành và nộp bản Đồ án tốt nghiệp cho Hội đồng thi.

Ngày.....tháng.....năm 202...

Sinh viên làm Đồ án tốt nghiệp
(Ký và ghi rõ Họ tên)

Phạm Thanh Sơn

LỜI CAM ĐOAN

Tác giả xin cam đoan đây là Đồ án tốt nghiệp tốt nghiệp của bản thân tác giả. Các kết quả trong Đồ án tốt nghiệp tốt nghiệp này là trung thực, và không sao chép từ bất kỳ một nguồn nào và dưới bất kỳ hình thức nào. Việc tham khảo các nguồn tài liệu đã được thực hiện trích dẫn và ghi nguồn tài liệu tham khảo đúng quy định.

Tác giả ĐATN

Phạm Thanh Sơn

LỜI CÁM ƠN

Sau hơn 4 năm học tập và nghiên cứu tại Khoa Công nghệ thông tin - Trường Đại học Thủy Lợi, em đã được trải nghiệm trong môi trường đào tạo tốt và nhận được sự chỉ dạy nhiệt tình của các thầy, các cô trong khoa.

Trước hết, em xin được bày tỏ lòng biết ơn sâu sắc nhất tới Thầy giáo hướng dẫn **PGS.TS. Lê Đức Hậu** đã tận tình giúp đỡ, hướng dẫn em rất nhiều trong suốt quá trình tìm hiểu và nghiên cứu và hoàn thành báo cáo tốt nghiệp.

Em cũng xin chân thành cảm ơn các thầy cô giáo trong khoa Công nghệ thông tin nói riêng và trường Đại học Thủy Lợi nói chung đã trang bị cho em những kiến thức cơ bản cần thiết làm hành trang trong những năm học tập tại trường.

Ngoài ra, em cũng chân thành cảm ơn đến bạn bè, các anh, các chị đã tận tình giúp đỡ, chia sẻ kiến thức, kinh nghiệm cũng như chỉ bảo em những thiếu sót trong quá trình học tập và nghiên cứu.

Cuối cùng em xin bày tỏ lòng biết ơn sâu sắc đến cha mẹ và những người thân trong gia đình đã dành cho em sự quan tâm đặc biệt, luôn hỗ trợ, động viên và tạo mọi điều kiện thuận lợi nhất cho em trong suốt thời gian qua và đặc biệt trong thời gian em làm đồ án tốt nghiệp.

Trong quá trình nghiên cứu của mình, mặc dù được sự hướng dẫn rất nhiệt tình, nghiêm túc của **PGS.TS. Lê Đức Hậu** cùng với sự nỗ lực của cá nhân nhưng cũng không thể tránh được những thiếu sót. Em rất mong nhận được sự đóng góp ý kiến của tất cả các quý Thầy, Cô giáo và các bạn để đề tài được hoàn thiện.

Tác giả ĐATN

Phạm Thanh Sơn



BẢN TÓM TẮT ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP

TÊN ĐỀ TÀI: Phát triển game The Knight Return

Sinh viên thực hiện: Phạm Thanh Sơn

Lớp: 62TH-NB

Giáo viên hướng dẫn: PGS.TS Lê Đức Hậu

TÓM TẮT ĐỀ TÀI

1. Bài toán cần giải quyết

Công nghệ thông tin ngày càng thể hiện được vai trò to lớn và hết sức quan trọng đối với toàn bộ đời sống con người. Cùng với đó, ngành công nghiệp game trên thế giới nói chung và khu vực Đông Nam Á nói riêng đang phát triển mạnh mẽ. Với sự bùng nổ của công nghệ di động và Internet, game trở thành một phần quan trọng của giải trí số và có sức ảnh hưởng rộng lớn. Việc nghiên cứu và phát triển game không chỉ mở ra nhiều cơ hội nghề nghiệp mà còn giúp sinh viên nắm bắt kịp xu hướng phát triển của thế giới. Giải pháp công nghệ. Làm game là một lĩnh vực kết hợp hoàn hảo giữa nghệ thuật và công nghệ. Sinh viên sẽ có cơ hội phát huy khả năng sáng tạo, đồng thời nâng cao kỹ năng kỹ thuật qua việc lập trình, thiết kế đồ họa, âm thanh và trải nghiệm người dùng. Điều này không chỉ giúp họ hoàn thiện bản thân mà còn tạo ra những sản phẩm có giá trị cao. Trong cuộc sống hiện đại đầy áp lực, mọi người thường tìm đến game như một hình thức giải trí để thư giãn và giảm căng thẳng. Game mang lại niềm vui và tạo ra những khoảnh khắc giải trí bổ ích, giúp người chơi tạm quên đi những lo toan hàng ngày. Vì vậy, em quyết định chọn đề tài với tên đề tài: “Phát triển game The Knight Return”.

1. Giải pháp công nghệ

- Công nghệ Unity
- Ngôn ngữ lập trình và các framework:
 - + Unity C#

+ Unity framework.

CÁC MỤC TIÊU CHÍNH

Vận dụng kiến thức đã học để phân tích và thiết kế một trò chơi điện tử có cốt truyện và hệ thống nhiệm vụ rõ ràng.

Vận dụng kiến thức về Unity và các thư viện, framework hỗ trợ để phát triển trò chơi.

Tiếp cận cách triển khai trò chơi Unity trên nền tảng window và quản lý dữ liệu người chơi bằng JSON Serialization lưu và tải các đối tượng phức tạp bằng cách sử dụng serialization. Unity hỗ trợ JSON và XML để serialize và deserialize các đối tượng..

Phát triển kỹ năng xây dựng và triển khai trò chơi trên nền tảng window

Rèn luyện kỹ năng: Khảo sát, phân tích thiết kế hệ thống trò chơi, thiết kế cơ sở dữ liệu, lập trình trò chơi bằng Unity

KẾT QUẢ DỰ KIẾN

Thiết kế, xây dựng được Game Play với các chức năng cơ bản để người chơi có thể trải nghiệm:

+ Di chuyển nhân vật.

+ Yếu tố hành động như tấn công quái vật.

+ Khám phá bản đồ vượt qua các chướng ngại vật.

+ Thu thập vàng để mua đồ nâng nhân vật

Thiết kế âm thanh và giao diện người chơi đẹp mắt

Lưu trữ và quản lý dữ liệu người chơi.

MỤC LỤC

BẢN TÓM TẮT ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP	iii
DANH MỤC KÝ HIỆU VÀ VIẾT TẮT	x
CHƯƠNG 1 CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ PHÁT TRIỂN GAME ..	1
1.1 Bài toán về công nghiệp trò chơi điện tử	1
1.1.1 Mô tả bài toán.....	1
1.1.3 Giải pháp thực hiện	2
1.2 Tổng quan về game engine	3
1.2.1 Định nghĩa Game Engine.....	3
1.2.2 Thành phần cơ bản của Game Engine	4
1.2.3 Lợi ích của việc sử dụng Game Engine	4
1.2.4 Một số Game Engine phổ biến	5
1.3 Công nghệ phát triển Unity Engine	5
1.3.1 Tổng quan về engine Unity.....	6
1.3.2 Tổng quan về C# trong Unity	7
1.3.3 Các Unity Framework.....	9
1.3.4 JSON Serialization và Deserialization trong Unity	12
1.3.5 Quy trình phát triển game bằng Unity	13
CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	16
2.1 Phân tích hệ thống về chức năng	16
2.1.1 Quy tắc nghiệp vụ hệ thống	16
2.1.2 Xác định và mô tả các tác nhân	16
2.1.3 Xác định và mô tả các ca sử dụng	17
2.1.4 Xây dựng biểu đồ ca sử dụng	18
2.1.5 Xây dựng kịch bản	23
2.2 Hệ thống quản lý dữ liệu	33
2.2.1 Lưu trữ và tái lại dữ liệu	33
2.2.2 Mã hóa dữ liệu	34
CHƯƠNG 3: TỔNG QUAN VỀ GAME “THE KNIGHT RETURN”	35
3.1 Giới thiệu về game	35
3.1.1 Thể loại game.....	35
3.1.2 Yếu tố	35

3.1.3 Nội dung.....	35
3.1.4 Chủ đề	36
3.1.5 Phong cách.....	36
3.1.6 Đối tượng người chơi.....	36
3.1.7 Lối chơi	36
3.1.8. Giao diện và cảm nhận.....	36
3.1.9 Khía cạnh tác động đến người chơi	37
3.1.10 Mục tiêu trải nghiệm	37
3.2 Lối chơi và cách vận hành trò chơi.....	37
3.2.1. Lối chơi	37
3.2.2 Nhân vật	37
3.2.3 Quái vật và bẫy	39
3.2.4 Cửa hàng và kho đồ	41
3.2.5 Level design	43
3.2.6 Game mode	44
3.3. Game Control	62
3.4. Thắng và thua.....	62
CHƯƠNG 4: CÀI ĐẶT TRÒ CHƠI VÀ KẾT LUẬN.....	63
4.1 Môi trường cài đặt.....	63
4.1.1 Unity Hub.....	63
4.1.2 Unity Editor	64
4.1.3 Visual Studio.....	66
4.2 Một số kết quả đạt được	67
4.2.1 Giao diện tùy chọn của trò chơi	67
4.2.1 Giao diện của trò chơi.....	69
4.3 Kết luận và hướng phát triển.....	73
4.3.1 Mục tiêu đạt được	73
4.3.2 Hạn chế	73
4.4 Hướng phát triển	73
TÀI LIỆU THAM KHẢO	74

DANH MỤC HÌNH ẢNH

Hình 2.1 Biểu đồ phân rã usecase tổng quát	18
Hình 2.2 Biểu đồ phân rã usecase chức năng chơi trò chơi	19
Hình 2.3 Biểu đồ phân rã usecase chức năng chỉnh trò chơi	19
Hình 2.4 Biểu đồ phân rã usecase chức năng điều khiển nhân vật	19
Hình 2.5 Biểu đồ phân rã usecase chức năng giao tranh với quái vật.....	20
Hình 2.6 Biểu đồ phân rã usecase chức năng thu thập vật phẩm.....	20
Hình 2.7 Biểu đồ phân rã usecase chức năng tương tác với NPCs	20
Hình 2.8 Biểu đồ phân rã usecase chức năng tìm kiếm kĩ năng.....	21
Hình 2.9 Biểu đồ phân rã usecase chức năng tạm dừng trò chơi	21
Hình 2.10 Biểu đồ phân rã usecase chức năng truy cập kho đồ	21
Hình 2.11 Biểu đồ phân rã usecase chức năng tương tác với NPCs của NPCs	22
Hình 2.12 Biểu đồ phân rã usecase chức năng tìm kiếm kĩ năng của NPCs.....	22
Hình 2.13 Biểu đồ phân rã usecase chức năng giao tranh với quái vật của kẻ địch	22
Hình 2.14 Biểu đồ hoạt động chức năng khởi động game	23
Hình 2.15 Biểu đồ hoạt động chức năng thoát game	24
Hình 2.16 Biểu đồ hoạt động chức năng điều khiển nhân vật.....	25
Hình 2.17 Biểu đồ hoạt động chức sử dụng linh hồn để hồi máu	26
Hình 2.18 Biểu đồ hoạt động chức năng giao tranh với quái vật.....	27
Hình 2.19 Biểu đồ hoạt động chức năng thu thập vật phẩm	28
Hình 2.20 Biểu đồ hoạt động chức năng tương tác với NPCs	29
Hình 2.21 Biểu đồ hoạt động chức năng tìm kiếm kĩ năng.....	30
Hình 2.22 Biểu đồ hoạt động chức năng truy cập kho đồ	31
Hình 2.23 Biểu đồ hoạt động chức năng tạm dừng trò chơi	32
Hình 2.23 Chức năng lưu trữ dữ liệu.....	33
Hình 2.24 Chức năng tải lên dữ liệu.....	34
Hình 2.25 Chức năng mã hóa dữ liệu	34
Hình 2.26 Vị trí file lưu trữ dữ liệu	34
Hình 3.1 Điều khiển nhân vật.....	38
Hình 3.2 Hướng dẫn sử dụng kĩ năng.....	38
Hình 3.3 UI thanh máu, linh hồn, vàng	39
Hình 3.4 Điểm hồi sinh	40
Hình 3.5 Cạm bẫy	40
Hình 3.6 Cửa hàng bán trang bị.....	41
Hình 3.7 Kho đồ	41

Hình 3.8 Cửa hàng bán minimap.....	42
Hình 3.9 Minimap	42
Hình 3.10 Map 1: Dark Forest.....	43
Hình 3.11 Map 2: Mystic Cave	43
Hình 3.12 Map 3: Green Forest.....	43
Hình 3.13 Map 4: Endless Sands.....	44
Hình 3.14 Map 5: Devil's Castle.....	44
Hình 3.15 Worm	45
Hình 3.16 Fast Worm	46
Hình 3.17 Around Worm.....	46
Hình 3.18 Flying.....	46
Hình 3.19 Crazy Flying	47
Hình 3.20 BlueBirdAngry	48
Hình 3.21 MadHog	48
Hình 3.22 Walk Tree	50
Hình 3.23 Jump Frog	50
Hình 3.24 Flying Frog	50
Hình 3.25 Boom Worm	51
Hình 3.26 Explosion Flower.....	51
Hình 3.27 Mom Butterfly & Litter Butterfly	51
Hình 3.28 Platform Worm	52
Hình 3.29 Fairy Poison.....	53
Hình 3.30 Tusk	53
Hình 3.31 DogMachine	54
Hình 3.32 Lamp Machine.....	55
Hình 3.33 OneEyeMachine	55
Hình 3.34 BigEyeMachine	55
Hình 3.35 RaMachine.....	57
Hình 3.36 NuclearMachine	57
Hình 3.37 FireDevil.....	58
Hình 3.38 FlyDevil.....	59
Hình 3.39 DevilHog	59
Hình 3.40 JumpDevil.....	59
Hình 3.41 WallWorm	60
Hình 3.42 PhoenixMachine	61

Hình 3.43 Brings Of Death.....	61
Hình 3.44 Cổng kết thúc game	62
Hình 4.1 Menu chính	67
Hình 4.2 Menu tùy chỉnh âm thanh	68
Hình 4.3 Menu chọn lưu trữ	68
Hình 4.4 Giao diện kết thúc trò chơi	69
Hình 4.5 Hình ảnh bản đồ.....	69
Hình 4.6 Giao tranh với quái vật	70
Hình 4.7 Vượt qua cạm bẫy.....	70
Hình 4.8 Né tránh đòn tấn công của quái vật	71
Hình 4.9 CheckPoint và SkillPoint.....	71
Hình 4.10 Bẫy lưỡi cưa	72
Hình 4.10 Menu dừng trò chơi	72

DANH MỤC BẢNG BIỂU

Bảng 2.1 Bảng mô tả các tác nhân.....	16
Bảng 2.2 Bảng kịch bản khởi động game	23
Bảng 2.3 Bảng kịch bản thoát game	24
Bảng 2.4 Bảng kịch bản điều khiển nhân vật	25
Bảng 2.5 Bảng kịch bản sử dụng linh hồn để hồi máu	26
Bảng 2.6 Bảng kịch bản giao tranh với quái vật	27
Bảng 2.7 Bảng kịch bản thu thập vật phẩm.....	28
Bảng 2.8 Bảng kịch bản tương tác với NPCs	29
Bảng 2.9 Bảng kịch bản tìm kiếm kỹ năng	30
Bảng 2.10 Bảng kịch bản truy cập kho đồ.....	31
Bảng 2.11 Bảng kịch bản tạm dừng trò chơi	32
Bảng 3.1 Bảng quái vật thường map Dark Forest, Mystic Cave	45
Bảng 3.2 Quái vật boss map Dark Forest, Mystic Cave.....	47
Bảng 3.3 Bảng quái vật thường map Green Forest	49
Bảng 3.4 Quái vật boss map Green Forest	52
Bảng 3.5 Bảng quái vật thường map Endless Sands	54
Bảng 3.6 Quái vật boss map Endless Sands	56
Bảng 3.7 Bảng quái vật thường map Devil's Castle	58
Bảng 3.8 Quái vật boss map Devil's Castle	60

DANH MỤC KÝ HIỆU VÀ VIẾT TẮT

Tùy viết tắt	Tên đầy đủ	Giải thích
API	Application Programming Interface	Là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác.
CPU	Central Processing Unit	Bộ xử lý trung tâm, là phần chính của máy tính thực hiện các lệnh của chương trình.
ĐATN	Đồ án tốt nghiệp	Đồ án tốt nghiệp.
NPC	Non-player character	Là một nhân vật trong các trò chơi được điều khiển bởi máy tính.
JSON	JavaScript Object Notation	Định dạng dữ liệu nhẹ sử dụng để trao đổi dữ liệu giữa các hệ thống và ứng dụng trong mạng.
RAM	Random Access Memory	Bộ nhớ truy cập ngẫu nhiên, là một loại bộ nhớ máy tính có thể truy cập dữ liệu một cách ngẫu nhiên.

CHƯƠNG 1 CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ PHÁT TRIỂN GAME

1.1 Bài toán về công nghiệp trò chơi điện tử

1.1.1 Mô tả bài toán

Công nghệ thông tin ngày càng thể hiện được vai trò to lớn và hết sức quan trọng đối với toàn bộ đời sống con người. Và ngành công nghiệp game trên thế giới nói chung và khu vực Đông Nam Á nói riêng đang phát triển mạnh mẽ. Với sự bùng nổ của công nghệ di động và Internet, game trở thành một phần quan trọng của giải trí số và có sức ảnh hưởng rộng lớn. Việc nghiên cứu và phát triển game không chỉ mở ra nhiều cơ hội nghề nghiệp mà còn giúp sinh viên nắm bắt kịp xu hướng phát triển của thế giới.

Game, hay trò chơi, là các hoạt động giải trí được thiết kế để tạo ra niềm vui, thử thách, và sự tương tác giữa người chơi. Trò chơi điện tử, một dạng phổ biến của game, là các trò chơi được thực hiện trên các thiết bị điện tử như máy tính, console, điện thoại di động, và các thiết bị chơi game chuyên dụng. Trò chơi điện tử thường bao gồm nhiều thể loại khác nhau như hành động, phiêu lưu, chiến lược, thể thao, và mô phỏng, đáp ứng nhu cầu giải trí đa dạng của người chơi. Trò chơi điện tử cung cấp một phương tiện giải trí phong phú và đa dạng, giúp người chơi giảm căng thẳng và thư giãn sau những giờ làm việc và học tập căng thẳng. Với sự phát triển của công nghệ, trò chơi điện tử ngày càng trở nên hấp dẫn và phong phú về nội dung, mang lại những trải nghiệm giải trí sống động và thú vị. Chơi game không chỉ giúp giải trí mà còn giúp phát triển nhiều kỹ năng quan trọng. Ví dụ, các trò chơi chiến lược giúp cải thiện khả năng lập kế hoạch và quản lý tài nguyên, trong khi các trò chơi hành động nhanh giúp tăng cường phản xạ và khả năng tập trung. Nhiều nghiên cứu đã chỉ ra rằng trò chơi điện tử có thể cải thiện kỹ năng giải quyết vấn đề, tư duy logic và sự sáng tạo của người chơi.

Chính vì vậy, em quyết định chọn đề tài: "Phát triển game The Knight Return". Việc phát triển "The Knight Return" không chỉ là một dự án kỹ thuật mà còn là một cơ hội để em khám phá và thể hiện sự sáng tạo trong thiết kế trò chơi. Qua đề tài này, em hi vọng sẽ góp phần vào sự phát triển của ngành công nghiệp game, đồng thời mang lại những

trải nghiệm giải trí tuyệt vời cho người chơi.

1.1.2 Yêu cầu thực tế

Để hoàn thiện đề tài: “Phát triển game The Knight Return” thì yêu cầu đặt ra không hề đơn giản. Dưới đây là một số yêu cầu khi thiết kế:

- Thiết kế giao diện game đơn giản dễ dàng sử dụng.
- Cách chơi cơ bản dễ dàng tiếp cận.
- Hệ thống điều khiển mượt mà, cấp độ khó tăng dần theo thời gian.
- Game có cốt truyện, nội dung thú vị gây hứng thú với người chơi.
- Tương thích trên nền tảng window đảm bảo hiệu suất tối ưu.
- Game yêu cầu bảo mật dữ liệu tránh người dùng sửa đổi.
- Hệ thống Save & Load lưu trữ tiến trình của người chơi

1.1.3 Giải pháp thực hiện

1.1.3.1 Thiết kế giao diện game đơn giản dễ dàng sử dụng

Giao diện người dùng (UI): Giao diện người dùng cần được thiết kế trực quan, dễ dàng điều hướng. Các menu, nút bấm và biểu tượng phải rõ ràng và dễ hiểu, giúp người chơi nhanh chóng nắm bắt được cách sử dụng mà không cần phải tìm hiểu quá nhiều.

Hướng dẫn người chơi: Cung cấp các hướng dẫn cụ thể và dễ hiểu trong quá trình chơi, giúp người chơi mới có thể làm quen với cơ chế và cách thức chơi game một cách nhanh chóng. Điều này có thể bao gồm các hướng dẫn bằng hình ảnh, video ngắn hoặc các chỉ dẫn trực tiếp trong game.

1.1.3.2 Game Play cơ bản dễ dàng tiếp cận

Hệ thống điều khiển: Cơ chế điều khiển cần mượt mà, chính xác và dễ dàng tiếp cận cho người chơi ở mọi cấp độ. Đối với phiên bản trên PC, cần hỗ trợ của cả bàn phím và chuột. Đối với phiên bản trên console hoặc thiết bị di động, cần tối ưu hóa cho các phím bấm hoặc màn hình cảm ứng.

Cấp độ khó phù hợp: Game nên có các mức độ khó khác nhau, từ dễ đến khó, để người chơi có thể chọn lựa phù hợp với trình độ của mình. Điều này giúp game có thể tiếp cận

được đông đảo người chơi với nhiều trình độ khác nhau. Cấp độ khó sẽ tăng dần theo thời gian, giúp người chơi cảm thấy được thử thách và không nhàm chán.

1.1.3.3 Game có cốt truyện, nội dung thú vị gây hứng thú với người chơi

Xây dựng cốt truyện hấp dẫn: Phát triển một cốt truyện có chiều sâu với các tình tiết lôi cuốn, nhân vật phát triển và những bất ngờ thú vị. Cốt truyện cần được truyền tải thông qua các đoạn hội thoại, cảnh cắt và các nhiệm vụ trong game.

Nội dung phong phú: Tạo ra nhiều cấp độ và thử thách đa dạng để người chơi luôn cảm thấy mới mẻ và hứng thú khi chơi game. Mỗi nhiệm vụ và thử thách nên liên kết chặt chẽ với cốt truyện chính, giúp người chơi cảm nhận được sự tiến triển và thành tựu.

1.1.3.4. Tương thích trên nền tảng Windows đảm bảo hiệu suất tối ưu

Tối ưu hóa hiệu suất: Đảm bảo game chạy mượt mà trên các thiết bị với cấu hình khác nhau, từ những thiết bị cao cấp đến những thiết bị có cấu hình trung bình và thấp. Sử dụng các công cụ và kỹ thuật tối ưu hóa như giảm bớt các hiệu ứng không cần thiết, tối ưu hóa mã nguồn và sử dụng các công nghệ đồ họa phù hợp.

Kiểm tra và thử nghiệm: Thực hiện các quá trình kiểm tra và thử nghiệm kỹ lưỡng trên nhiều cấu hình phần cứng khác nhau để đảm bảo game hoạt động ổn định và không gặp lỗi.

1.1.3.5. Game yêu cầu bảo mật dữ liệu tránh người dùng sửa đổi

Bảo vệ dữ liệu người chơi: Áp dụng các biện pháp bảo mật để bảo vệ dữ liệu cá nhân và thông tin tiền độ chơi của người chơi. Sử dụng mã hóa dữ liệu và lưu trữ trên máy tính

1.2 Tổng quan về game engine

1.2.1 Định nghĩa Game Engine

Game Engine là một nền tảng phần mềm được thiết kế để đơn giản hóa việc phát triển trò chơi điện tử. Nó cung cấp một bộ công cụ và thư viện hỗ trợ các nhà phát triển tạo ra, quản lý và tối ưu hóa các yếu tố khác nhau của trò chơi, bao gồm đồ họa, âm thanh,

vật lý, và trí tuệ nhân tạo.

1.2.2 Thành phần cơ bản của Game Engine

Graphics Engine: Chịu trách nhiệm hiển thị đồ họa 2D hoặc 3D của trò chơi. Nó bao gồm các công cụ để tạo ra và quản lý các mô hình, kết cấu, và hiệu ứng ánh sáng.

Physics Engine: Quản lý các quy luật vật lý trong trò chơi, như va chạm, trọng lực, và chuyển động của các đối tượng.

Audio Engine: Xử lý âm thanh trong trò chơi, bao gồm âm nhạc nền, hiệu ứng âm thanh và âm thanh 3D.

Scripting Engine: Hỗ trợ viết mã lệnh điều khiển hành vi của các đối tượng trong trò chơi. Ngôn ngữ lập trình thường được sử dụng bao gồm C++, C#, Lua, và Python.

Animation Engine: Quản lý các chuyển động của nhân vật và đối tượng trong trò chơi, bao gồm cả hoạt cảnh (animation) và điều khiển khung hình (frame control).

Networking Engine: Hỗ trợ các tính năng chơi đa người và trực tuyến, bao gồm kết nối mạng, đồng bộ hóa dữ liệu và quản lý máy chủ.

AI Engine: Quản lý trí tuệ nhân tạo trong trò chơi, bao gồm các hành vi của NPC (Non-Player Characters) và các thuật toán học máy.

1.2.3 Lợi ích của việc sử dụng Game Engine

Tiết kiệm thời gian và công sức: Các nhà phát triển không cần phải viết lại từ đầu tất cả các thành phần cơ bản của trò chơi. Họ có thể tận dụng các công cụ và thư viện có sẵn trong Game Engine.

Khả năng tái sử dụng: Game Engine cho phép tái sử dụng mã nguồn và tài nguyên giữa các dự án khác nhau, giúp tối ưu hóa quá trình phát triển.

Hỗ trợ đa nền tảng: Nhiều Game Engine hỗ trợ xuất bản trò chơi trên nhiều nền tảng khác nhau như PC, console, di động và web.

Cộng đồng và tài liệu hỗ trợ: Nhiều Game Engine có cộng đồng người dùng lớn và tài

liệu phong phú, giúp dễ dàng học tập và giải quyết các vấn đề gặp phải trong quá trình phát triển.

1.2.4 Một số Game Engine phổ biến

Unity: Một trong những Game Engine phổ biến nhất, hỗ trợ đa nền tảng và được sử dụng rộng rãi trong cả phát triển trò chơi và ứng dụng tương tác khác.

Unreal Engine: Được phát triển bởi Epic Games, nổi tiếng với khả năng đồ họa cao cấp và thường được sử dụng trong các trò chơi AAA.

Godot: Một Game Engine mã nguồn mở, dễ học và sử dụng, thích hợp cho cả phát triển 2D và 3D.

Cry Engine: Nổi bật với đồ họa chất lượng cao, thường được sử dụng trong các trò chơi yêu cầu hiệu suất đồ họa mạnh mẽ.

RPG Maker: Chuyên dùng để phát triển các trò chơi nhập vai 2D, rất phổ biến trong cộng đồng phát triển indie.

Game Engine là một công cụ quan trọng và không thể thiếu trong việc phát triển trò chơi điện tử hiện đại. Nó giúp các nhà phát triển tiết kiệm thời gian, tối ưu hóa quy trình làm việc, và tập trung vào việc sáng tạo nội dung và cải thiện trải nghiệm người chơi. Với sự phát triển nhanh chóng của công nghệ, các Game Engine ngày càng mạnh mẽ và linh hoạt hơn, mở ra nhiều cơ hội cho cả các nhà phát triển chuyên nghiệp và những người mới bắt đầu.

1.3 Công nghệ phát triển Unity Engine

Trong thời đại công nghệ thông tin hiện nay, các sản phẩm công nghệ ngày càng bị đánh giá kỹ lưỡng hơn bởi người dùng, đặc biệt là các sản phẩm Game, thường nhận được nhiều ý kiến phản hồi từ cả game thủ lẫn những người chơi thông thường. Ngành công nghiệp Game đang chứng kiến sự bùng nổ với tốc độ phát triển nhanh chóng, hàng loạt các tựa game hấp dẫn đã ra mắt trong thời gian qua. Đằng sau sự thành công của những tựa game phát triển và nổi tiếng này đều có sự hỗ trợ của các Game Engine. Game Engine là một công cụ hỗ trợ, một Middleware giúp nhà phát triển viết game nhanh chóng và đơn giản hơn, đồng thời cung cấp khả năng tái sử dụng tài nguyên và mã nguồn

cao, cho phép phát triển nhiều tựa game từ một Game Engine.

Đồ án này sẽ khảo sát và nghiên cứu về Unity Engine - một Game Engine rất phổ biến và mạnh mẽ hiện nay, nhằm thực hiện việc phát triển một trò chơi (Demo) bắn máy bay 2D. Mục đích là trang bị kiến thức và kỹ năng cho định hướng nghề nghiệp phát triển game của chúng em, đóng góp vào sự phát triển của ngành công nghiệp Game trong nước. Các chương đầu của tài liệu sẽ lần lượt trình bày các khái niệm chung về Game Engine và thế giới 2D.

1.3.1 Tổng quan về engine Unity.

Unity là một Game Engine được phát triển bởi Unity Technologies, được ra mắt lần đầu vào năm 2005. Unity hỗ trợ lập trình đa nền tảng, cho phép các nhà phát triển tạo ra các trò chơi và ứng dụng tương tác không chỉ trên máy tính để bàn mà còn trên các thiết bị di động, máy chơi game, và các nền tảng thực tế ảo (VR) và thực tế tăng cường (AR). Sau đây là những tính năng chính của engine Unity:

Đa nền tảng: Unity hỗ trợ phát triển game cho nhiều nền tảng khác nhau, bao gồm Windows, macOS, Linux, iOS, Android, WebGL, và các hệ máy console như PlayStation, Xbox và Nintendo Switch. Điều này giúp game "The Knight Return" có thể tiếp cận được đa dạng người chơi trên nhiều thiết bị khác nhau.

Giao diện trực quan: Unity cung cấp một giao diện người dùng (UI) trực quan và dễ sử dụng, với các công cụ kéo-thả giúp nhà phát triển dễ dàng tạo ra và chỉnh sửa các đối tượng trong game. Giao diện này giúp tối ưu hóa quy trình phát triển và giảm bớt thời gian học tập cho người mới.

Công cụ đồ họa mạnh mẽ: Unity hỗ trợ các công nghệ đồ họa tiên tiến như đồ họa 2D và 3D, ánh sáng động, đổ bóng, và hiệu ứng hậu kỳ (post-processing). Điều này giúp tạo ra hình ảnh chất lượng cao và trải nghiệm thị giác hấp dẫn cho người chơi.

Hệ thống vật lý và hoạt ảnh: Unity tích hợp sẵn hệ thống vật lý mạnh mẽ và công cụ tạo hoạt ảnh (animation), giúp tạo ra các hiệu ứng chuyển động và tương tác vật lý chân thực trong game. Các công cụ này rất hữu ích cho việc phát triển cơ chế điều khiển và hoạt động của nhân vật trong "The Knight Return".

Hệ thống script: Unity sử dụng ngôn ngữ lập trình C# cho việc viết script, giúp nhà phát triển dễ dàng tạo ra các tính năng và logic cho game. C# là ngôn ngữ phổ biến và mạnh mẽ, hỗ trợ phát triển game một cách hiệu quả.

Unity Editor: Đây là công cụ chính của Unity, nơi người dùng có thể thiết kế, lập trình và kiểm tra trò chơi. Unity Editor bao gồm nhiều cửa sổ và công cụ hỗ trợ cho việc chỉnh sửa scene, quản lý asset, và lập trình.

Cộng đồng và tài liệu phong phú: Unity có một cộng đồng nhà phát triển lớn và sôi động, cùng với tài liệu hướng dẫn chi tiết và nhiều tài nguyên học tập. Nhà phát triển có thể dễ dàng tìm thấy các giải pháp cho vấn đề của mình hoặc học hỏi kinh nghiệm từ cộng đồng.

Asset Store: Unity Asset Store cung cấp một kho tài nguyên phong phú, bao gồm các mẫu đồ họa, âm thanh, script, và các plugin bổ sung. Nhà phát triển có thể tận dụng các tài nguyên này để tiết kiệm thời gian và công sức trong quá trình phát triển game.

Unity là một công cụ phát triển game mạnh mẽ và đa năng, mang lại nhiều lợi ích cho nhà phát triển game từ cá nhân đến các studio lớn. Với các tính năng mạnh mẽ, hỗ trợ đa nền tảng, và cộng đồng phát triển sôi động, Unity là lựa chọn hàng đầu cho việc phát triển game "The Knight Return" và nhiều dự án game khác. Việc sử dụng Unity không chỉ giúp tối ưu hóa quy trình phát triển mà còn mang lại trải nghiệm tuyệt vời cho người chơi.

1.3.2 Tổng quan về C# trong Unity

1.3.2.1. Giới thiệu về C# trong Unity

C# là ngôn ngữ lập trình chính được sử dụng trong Unity, một engine phát triển game phổ biến. Unity hỗ trợ C# vì tính mạnh mẽ, hiệu suất cao và cú pháp dễ hiểu, phù hợp cho việc phát triển game. C# được phát triển bởi Microsoft và là một phần của nền tảng .NET, giúp nó có một cộng đồng phát triển lớn và nhiều tài nguyên hỗ trợ.

1.3.2.2. Lý do chọn C# cho Unity

Tính mạnh mẽ và hiệu suất cao: C# cung cấp các tính năng lập trình hiện đại như lambda expressions, LINQ, và async/await, giúp viết mã nguồn hiệu quả và tối ưu. Điều này đặc

biệt quan trọng trong phát triển game, nơi hiệu suất là yếu tố then chốt.

Cú pháp dễ hiểu: C# có cú pháp dễ học và dễ hiểu, đặc biệt với những người đã quen thuộc với các ngôn ngữ lập trình hướng đối tượng khác như Java hoặc C++. Điều này giúp các nhà phát triển nhanh chóng làm quen và bắt đầu phát triển game trong Unity.

Tích hợp tốt với Unity: Unity cung cấp các API và thư viện phong phú bằng C#, giúp dễ dàng tạo và quản lý các thành phần trong game. Sự tích hợp chặt chẽ giữa Unity và C# giúp tối ưu hóa quy trình phát triển và giảm bớt sự phức tạp.

Tài liệu và cộng đồng hỗ trợ: C# có một cộng đồng lớn và nhiều tài liệu hướng dẫn, từ các khóa học trực tuyến đến các diễn đàn hỗ trợ. Nhà phát triển có thể dễ dàng tìm thấy giải pháp cho các vấn đề của mình và học hỏi kinh nghiệm từ cộng đồng.

1.3.2.3. Các thành phần chính trong lập trình C# cho Unity

Script: Scripts trong Unity được viết bằng C# để định nghĩa hành vi và logic cho các đối tượng trong game. Một script cơ bản thường bao gồm các phương thức như Start(), Update(), và FixedUpdate() để quản lý các sự kiện trong vòng đời của đối tượng.

GameObject và Component: Trong Unity, mọi thứ đều là một GameObject, và các Component được gắn vào GameObject để thêm chức năng. Các script C# thường được gắn vào GameObject dưới dạng Component, giúp dễ dàng quản lý và tổ chức mã nguồn.

API và Thư viện Unity: Unity cung cấp một bộ API và thư viện phong phú cho việc quản lý các yếu tố như vật lý, âm thanh, hình ảnh, và đầu vào. Các API này giúp nhà phát triển dễ dàng tạo ra các hiệu ứng và tương tác phức tạp trong game.

Sự kiện và Hệ thống sự kiện: Unity sử dụng hệ thống sự kiện để quản lý các tương tác giữa người chơi và game. Các sự kiện như nhấp chuột, va chạm, và kích hoạt trigger đều có thể được xử lý bằng các script C#, giúp tạo ra trải nghiệm chơi game động và phản hồi nhanh.

C# là ngôn ngữ lập trình mạnh mẽ và hiệu quả, được sử dụng rộng rãi trong phát triển game với Unity. Với sự hỗ trợ từ Unity Editor, các API phong phú, và cộng đồng phát triển lớn, C# giúp nhà phát triển dễ dàng tạo ra các trò chơi chất lượng cao. Việc nắm

vững C# không chỉ giúp bạn phát triển game một cách hiệu quả mà còn mở rộng cơ hội nghề nghiệp trong lĩnh vực công nghệ thông tin và phát triển phần mềm.

1.3.3 Các Unity Framework

1.3.3.1. UnityEngine Framework

UnityEngine là nền tảng chính của Unity, cung cấp các lớp và API cần thiết để phát triển game. Nó bao gồm các module quan trọng sau:

Physics: Xử lý vật lý cho các đối tượng trong game, bao gồm va chạm, trọng lực, và lực.

Rendering: Quản lý việc hiển thị các đối tượng 2D và 3D trên màn hình.

Animation: Tạo và quản lý các hoạt ảnh của nhân vật và đối tượng.

Audio: Quản lý âm thanh, bao gồm phát và xử lý âm thanh.

Networking: Hỗ trợ tạo các ứng dụng mạng và game nhiều người chơi.

1.3.3.2. Unity Editor Framework

UnityEditor cung cấp các công cụ và API để mở rộng và tùy chỉnh Unity Editor. Các nhà phát triển có thể tạo ra các công cụ tùy chỉnh, menu, và cửa sổ riêng để tối ưu hóa quy trình phát triển.

Custom Editor Windows: Tạo các cửa sổ và công cụ tùy chỉnh trong Unity Editor.

Editor Scripting: Viết các script để tự động hóa các tác vụ trong Editor, như xây dựng asset bundles hoặc kiểm tra lỗi.

1.3.3.3. Unity UI (uGUI)

Unity UI là framework để tạo giao diện người dùng trong game. Nó bao gồm các công cụ để tạo và quản lý các phần tử UI như nút bấm, thanh trượt, và hộp thoại.

Canvas: Là nền tảng để vẽ các phần tử UI.

Event System: Quản lý các sự kiện tương tác người dùng như nhấp chuột và chạm màn hình.

Layout Groups: Sắp xếp và căn chỉnh các phần tử UI một cách tự động.

1.3.3.4. Unity 2D Framework

Unity 2D cung cấp các công cụ và API để phát triển game 2D, bao gồm:

Sprite Renderer: Hiển thị hình ảnh 2D (sprite) trên màn hình.

Tilemap: Tạo và quản lý bản đồ ô vuông (tile-based map) trong game.

Physics2D: Xử lý vật lý cho các đối tượng 2D.

1.3.3.5. Unity 3D Framework: Unity 3D cung cấp các công cụ và API để phát triển game 3D, bao gồm:

Mesh Renderer: Hiển thị các mô hình 3D trên màn hình.

Terrain: Tạo và quản lý địa hình 3D.

NavMesh: Quản lý điều hướng AI cho các đối tượng trong môi trường 3D.

1.3.3.6. Unity ECS (Entity Component System)

Unity ECS là một kiến trúc phát triển game hiệu suất cao, tách biệt dữ liệu và hành vi của các đối tượng game. ECS cho phép tối ưu hóa việc xử lý dữ liệu và cải thiện hiệu suất.

Entities: Các đơn vị dữ liệu cơ bản trong ECS.

Components: Chứa dữ liệu liên quan đến các entities.

Systems: Xử lý logic và hành vi dựa trên dữ liệu từ các components.

1.3.3.7. Unity Addressable Assets

Addressable Assets là hệ thống quản lý tài nguyên giúp dễ dàng tải và quản lý tài nguyên trong game, đặc biệt là các game có quy mô lớn hoặc nội dung động.

Addressable: Đánh dấu tài nguyên để quản lý chúng dễ dàng.

Profile: Quản lý các cấu hình tải tài nguyên khác nhau cho các nền tảng hoặc môi trường khác nhau.

1.3.3.8. Unity Timeline

Timeline là công cụ để tạo các đoạn cắt cảnh, hoạt cảnh phức tạp và điều khiển các sự kiện trong game.

Tracks: Các dòng thời gian chứa các đối tượng và sự kiện.

Clips: Các đoạn thời gian cụ thể chứa các hành động hoặc hoạt cảnh.

Signals: Các sự kiện kích hoạt trong quá trình chạy Timeline.

1.3.3.9. Unity Cinemachine

Cinemachine là công cụ quản lý camera mạnh mẽ, giúp tạo các chuyển động camera mượt mà và chuyên nghiệp mà không cần viết nhiều mã.

Virtual Cameras: Các camera ảo có thể được thiết lập và điều chỉnh tự động.

Cinemachine Brain: Quản lý các camera ảo và chuyển đổi giữa chúng một cách mượt mà.

Confiner: Giữ camera trong một khu vực giới hạn.

1.3.3.10. Unity Jobs System và Burst Compiler

Jobs System và Burst Compiler là các công cụ tối ưu hóa hiệu suất giúp tận dụng tối đa sức mạnh của CPU đa lõi.

Jobs System: Tạo và quản lý các công việc song song để tối ưu hóa hiệu suất.

Burst Compiler: Tăng hiệu suất của mã C# bằng cách biên dịch nó xuống mã máy được tối ưu hóa.

Unity cung cấp một loạt các framework mạnh mẽ giúp tối ưu hóa quy trình phát triển game và nâng cao chất lượng sản phẩm cuối cùng. Từ quản lý UI, đồ họa 2D/3D, hệ thống vật lý, đến tối ưu hóa hiệu suất, Unity giúp nhà phát triển tạo ra những trò chơi phong phú và hấp dẫn trên nhiều nền tảng khác nhau. Việc hiểu rõ và tận dụng các framework này sẽ giúp tối ưu hóa quy trình phát triển và nâng cao chất lượng game

1.3.4 JSON Serialization và Deserialization trong Unity

1.3.4.1. Giới thiệu về JSON

JSON (JavaScript Object Notation) là một định dạng trao đổi dữ liệu nhẹ, dễ đọc và dễ viết cho con người, đồng thời dễ dàng phân tích và sinh ra bởi máy móc. JSON thường được sử dụng để truyền dữ liệu giữa máy chủ và ứng dụng web, nhưng nó cũng phổ biến trong nhiều ứng dụng khác, bao gồm cả game development.

1.3.4.2. Tại sao sử dụng JSON trong Unity?

JSON được sử dụng rộng rãi trong Unity để lưu trữ và truyền tải dữ liệu, bao gồm cấu hình game, thông tin người chơi, trạng thái game và nhiều thứ khác. Lợi ích của việc sử dụng JSON bao gồm:

Định dạng nhẹ và dễ đọc: JSON dễ đọc và viết, làm cho việc kiểm tra và sửa lỗi trở nên dễ dàng hơn.

Khả năng tương thích tốt: JSON có thể được sử dụng trên nhiều nền tảng và ngôn ngữ lập trình khác nhau.

Tiện lợi cho việc lưu trữ và truyền tải dữ liệu: JSON là một lựa chọn tốt để lưu trữ dữ liệu tạm thời hoặc truyền tải dữ liệu qua mạng.

1.3.4.3. JSON Serialization và Deserialization trong Unity

Serialization là quá trình chuyển đổi đối tượng (object) thành một định dạng mà có thể lưu trữ hoặc truyền tải được, chẳng hạn như JSON. Deserialization là quá trình ngược lại, chuyển đổi dữ liệu từ định dạng JSON trở lại thành đối tượng.

Unity cung cấp một số phương pháp để thực hiện JSON Serialization và Deserialization như JsonUtility. Quy trình lưu trữ:

Unity C# script -> Json Utility -> String -> Disk và ngược lại để lấy dữ liệu.

1.3.4.4. Sử dụng JsonUtility trong Unity

JsonUtility là lớp được Unity cung cấp để làm việc với JSON. Nó nhanh chóng và dễ sử dụng cho các tác vụ đơn giản.

JsonUtility:

Ưu điểm:

Tích hợp sẵn trong Unity, không cần cài đặt thêm.

Hiệu suất cao cho các tác vụ đơn giản.

Nhược điểm:

Ít tính năng hơn, không hỗ trợ nhiều cấu trúc dữ liệu phức tạp.

JSON Serialization và Deserialization là các kỹ thuật quan trọng trong phát triển game bằng Unity, giúp quản lý và truyền tải dữ liệu một cách hiệu quả. Unity cung cấp sẵn JsonUtility để thực hiện các thao tác này một cách nhanh chóng và dễ dàng.

1.3.5 Quy trình phát triển game bằng Unity

Để hoàn thiện một dự án game trên Unity, cần tuân theo các bước quy trình phát triển sau:

1.3.5.1 Thiết kế và lập kế hoạch:

Xác định ý tưởng và khái niệm game: Lên ý tưởng và xác định mục tiêu của game, bao gồm cốt truyện, nhân vật, và các yếu tố lối chơi chính.

Xây dựng cốt truyện và nhân vật: Phát triển cốt truyện chi tiết và thiết kế các nhân vật chính trong game.

Thiết kế các cấp độ và nhiệm vụ: Lên kế hoạch cho các cấp độ và nhiệm vụ, đảm bảo rằng game có sự cân bằng giữa thử thách và niềm vui.

Lập kế hoạch phát triển chi tiết: Xác định các giai đoạn phát triển, phân chia công việc và đặt ra các mốc thời gian cụ thể.

1.3.5.2. Tạo dựng game trong Unity:

Thiết kế giao diện và các màn hình game: Sử dụng Unity Editor để thiết kế các màn hình menu, giao diện người dùng và các yếu tố đồ họa khác.

Xây dựng các mô hình 3D/2D và tạo hoạt ảnh: Tạo ra các mô hình và hoạt ảnh cho nhân vật, môi trường, và các đối tượng khác trong game.

Viết script cho các tính năng và logic game: Sử dụng C# để lập trình các tính năng, tương tác, và logic của game.

Tích hợp âm thanh và nhạc nền: Thêm các hiệu ứng âm thanh và nhạc nền để tăng cường trải nghiệm chơi game.

1.3.5.3. Kiểm tra và tối ưu hóa:

Thực hiện kiểm tra chất lượng và tìm lỗi: Kiểm tra game để phát hiện và sửa các lỗi, đảm bảo game hoạt động mượt mà và không có lỗi nghiêm trọng.

Tối ưu hóa hiệu suất và kích thước file: Tối ưu hóa mã nguồn, đồ họa và các tài nguyên khác để đảm bảo game chạy mượt mà trên các nền tảng khác nhau.

Điều chỉnh các yếu tố lối chơi và đồ họa: Dựa trên phản hồi từ thử nghiệm, điều chỉnh lối chơi và đồ họa để đảm bảo game hấp dẫn và cân bằng.

1.3.5.4. Phát hành và bảo trì:

Chuẩn bị và triển khai game trên các nền tảng đã chọn: Đóng gói và phát hành game trên các cửa hàng ứng dụng và nền tảng phân phối game.

Theo dõi phản hồi của người chơi và cập nhật game: Lắng nghe phản hồi từ người chơi và thực hiện các cập nhật cần thiết để cải thiện game.

Bảo trì và cải tiến game để duy trì sự hấp dẫn: Liên tục bảo trì và cập nhật game, thêm nội dung mới và cải thiện trải nghiệm chơi game.

1.3.5.5 Công cụ và tài nguyên bổ sung

Visual Studio: Visual Studio là môi trường phát triển tích hợp (IDE) phổ biến, được sử dụng để viết mã C# trong Unity. Nó cung cấp nhiều công cụ hỗ trợ lập trình như gỡ lỗi, quản lý mã nguồn, và tích hợp với Unity Editor.

Photoshop/GIMP: Photoshop và GIMP là các công cụ chỉnh sửa hình ảnh mạnh mẽ, giúp tạo ra các kết cấu và hình ảnh 2D cho game. Photoshop là một phần mềm thương mại, trong khi GIMP là một lựa chọn mã nguồn mở miễn phí.

Unity Collaborate và Git: Unity Collaborate và Git là các công cụ quản lý mã nguồn giúp quản lý và đồng bộ hóa mã nguồn và tài nguyên của dự án giữa các thành viên trong nhóm. Điều này rất hữu ích cho việc làm việc nhóm và theo dõi các thay đổi trong dự án.

CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1 Phân tích hệ thống về chức năng

2.1.1 Quy tắc nghiệp vụ hệ thống

2.1.1.1 Nhóm nghiệp vụ hệ thống

Người dùng có thể khởi động game, thoát game.

Các quyền cơ bản có trong hệ thống: Người chơi.

Người chơi sử dụng game bằng cách tải game.

2.1.1.2 Nhóm nghiệp vụ quản lý

Quản lý nhân vật: Nhân vật chính có thể nâng cấp sức mạnh, học thêm kỹ năng mới (ví dụ: Dash, Double Jump), trang bị để thay đổi cách chơi và cải thiện các thuộc tính.

Quản lý tài nguyên: Thu thập Gold (đơn vị tiền tệ trong trò chơi), sử dụng Gold để mua bản đồ, trang bị.

Quản lý nhiệm vụ: Nhiệm vụ chính là khám phá và tiêu diệt các boss.

Quản lý bản đồ và khu vực: Mở khóa các khu vực mới bằng cách khám phá và sử dụng các khả năng mới, các điểm lưu (checkpoint) nơi người chơi có thể nghỉ ngơi và lưu tiến độ.

Quản lý tương tác: Tương tác với các NPC mua đồ và thông tin, tương tác với môi trường để giải các câu đố, mở khóa đường đi.

2.1.2 Xác định và mô tả các tác nhân

Bảng 2.1 Bảng mô tả các tác nhân

Tác nhân	Mô tả
Người Chơi (Player)	Người chơi là tác nhân chính, điều khiển nhân vật chính của trò chơi, thực hiện các nhiệm vụ, chiến đấu với kẻ thù, khám phá thế giới và tương tác với các NPC.

Nhân Vật Điều Khiển Bởi Máy Tính (NPC)	NPC là các nhân vật không phải người chơi, được điều khiển bởi AI của trò chơi. Họ cung cấp thông tin, và dịch vụ khác nhau cho người chơi.
Kẻ Thủ (Enemies)	Kẻ thù là những sinh vật hoặc nhân vật đối kháng mà người chơi phải chiến đấu và tiêu diệt để tiến xa hơn trong trò chơi.

2.1.3 Xác định và mô tả các ca sử dụng

Chơi trò chơi: Người chơi bắt đầu và tham gia vào trò chơi từ màn hình chính.

Tùy chỉnh trò chơi: Người chơi có thể tùy chỉnh các thiết lập trong trò chơi như âm thanh

Thoát trò chơi: Người chơi có thể thoát khỏi trò chơi và trở về màn hình chính hoặc thoát hẳn khỏi ứng dụng.

Điều khiển nhân vật: Người chơi điều khiển nhân vật chính trong trò chơi.

Giao tranh với quái vật: Người chơi chiến đấu với các quái vật xuất hiện trong trò chơi.

Thu thập vật phẩm: Người chơi thu thập các vật phẩm trong trò chơi để sử dụng hoặc nâng cấp nhân vật.

Tương tác với NPCs: Người chơi tương tác với các nhân vật điều khiển bởi máy tính (NPC) để mua bán vật phẩm hoặc nhận thông tin.

Tìm kiếm kỹ năng: Người chơi khám phá và tìm kiếm các kỹ năng mới để nâng cao khả năng của nhân vật.

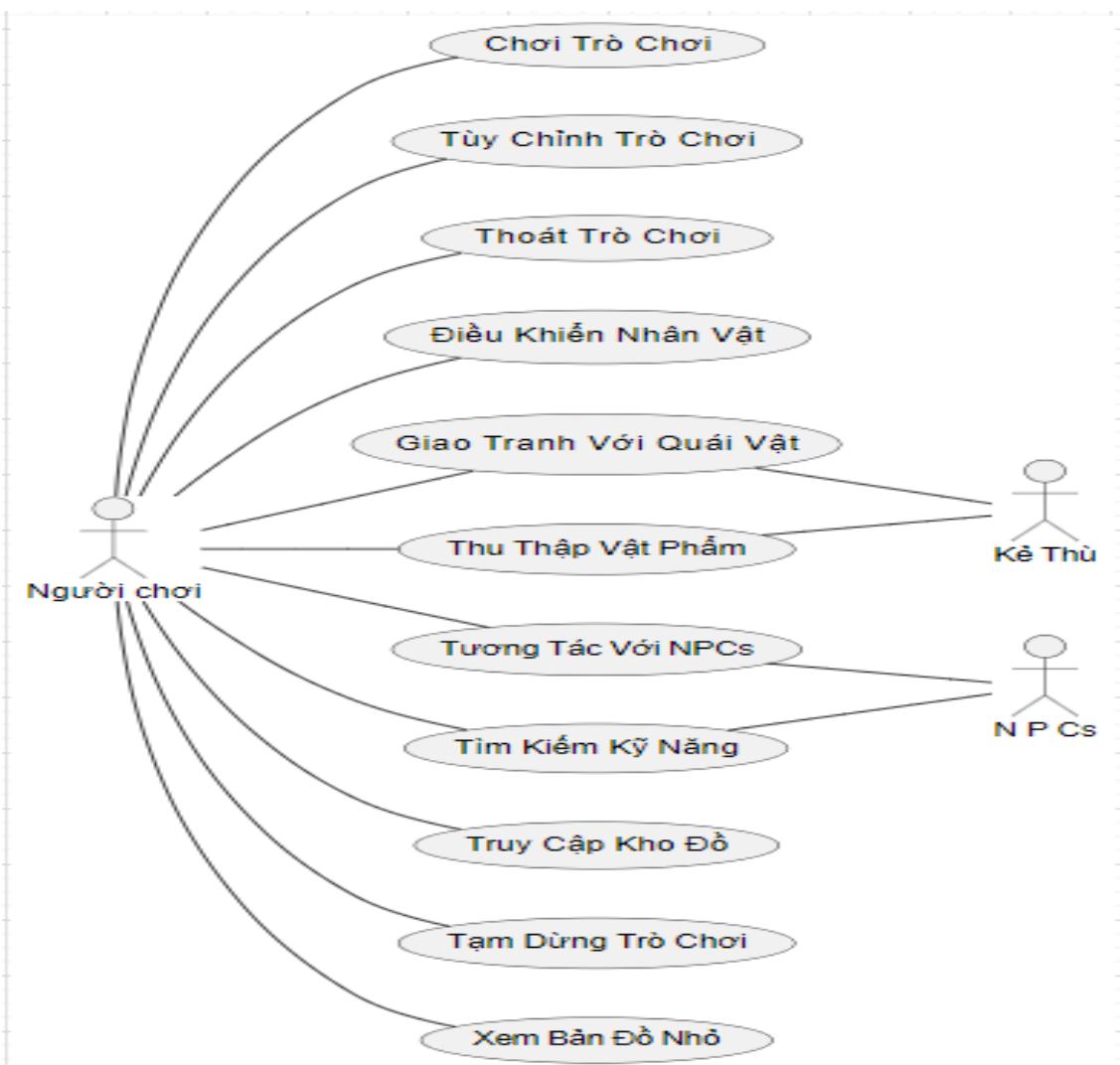
Truy cập kho đồ: Người chơi kiểm tra và quản lý các vật phẩm đã thu thập trong kho đồ.

Tạm dừng trò chơi: Người chơi có thể tạm dừng trò chơi để nghỉ ngơi hoặc truy cập các tùy chọn khác.

Xem bản đồ nhỏ: Người chơi có thể xem bản đồ nhỏ để biết vị trí hiện tại và các khu vực đã khám phá.

2.1.4 Xây dựng biểu đồ case sử dụng

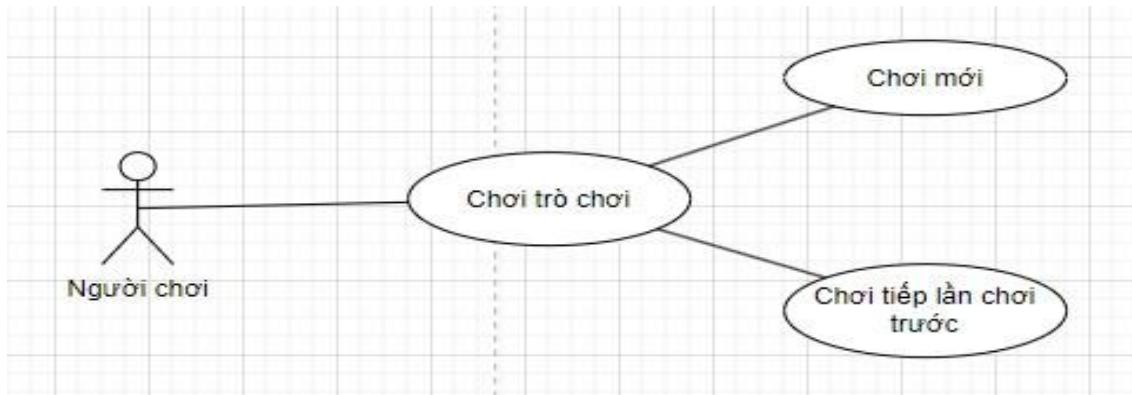
2.1.4.1 Biểu đồ usecase cho toàn hệ thống



Hình 2.1 Biểu đồ phân rã usecase tổng quát

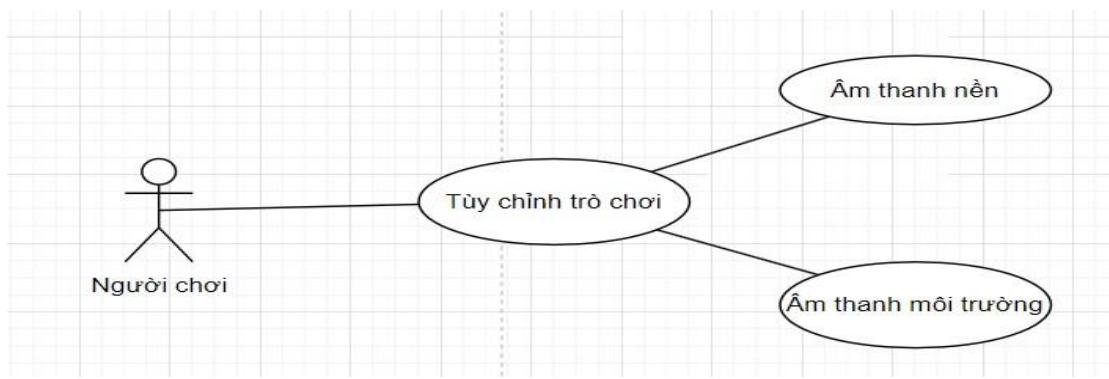
2.1.4.2 Biểu đồ phân rã usecase của tác nhân “Người chơi”

a) Chức năng chơi trò chơi



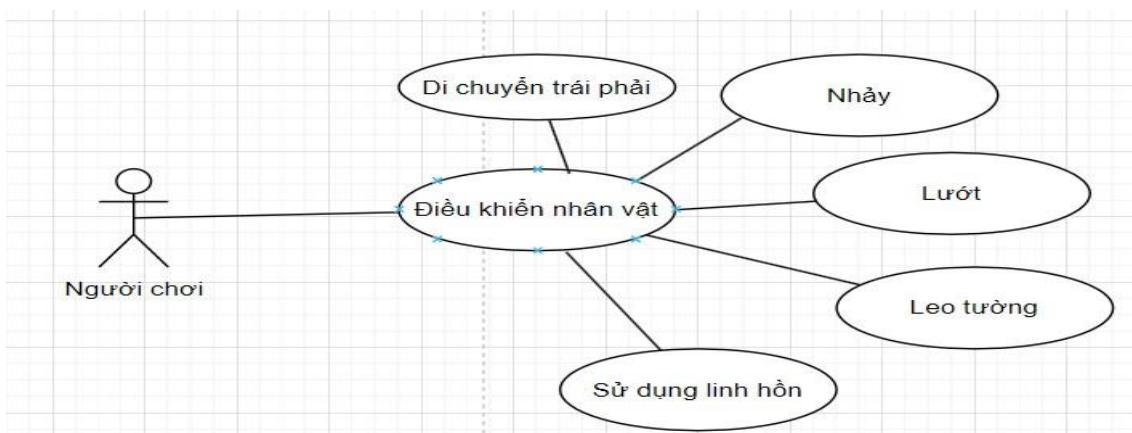
Hình 2.2 Biểu đồ phân rã usecase chức năng chơi trò chơi

b) Chức năng tùy chỉnh trò chơi



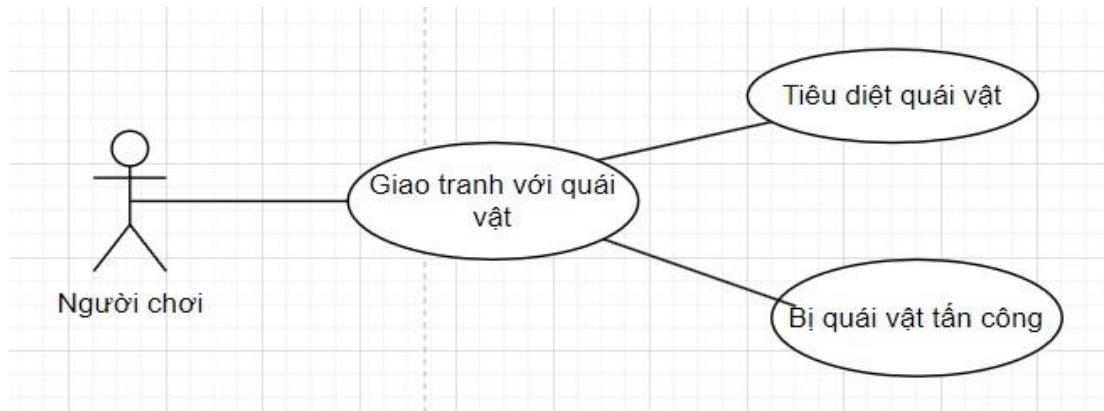
Hình 2.3 Biểu đồ phân rã usecase chức năng chỉnh trò chơi

c) Chức năng điều khiển nhân vật



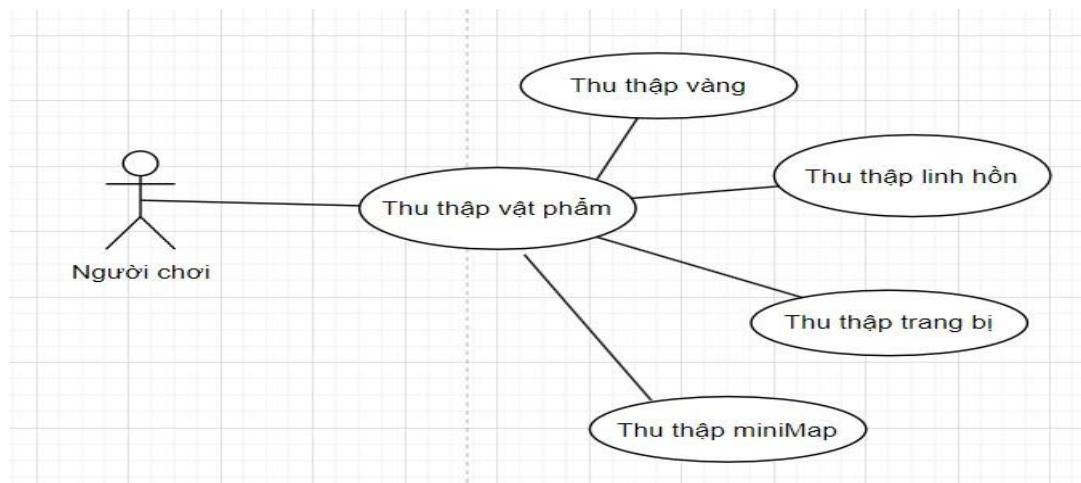
Hình 2.4 Biểu đồ phân rã usecase chức năng điều khiển nhân vật

d) Chức năng giao tranh với quái vật



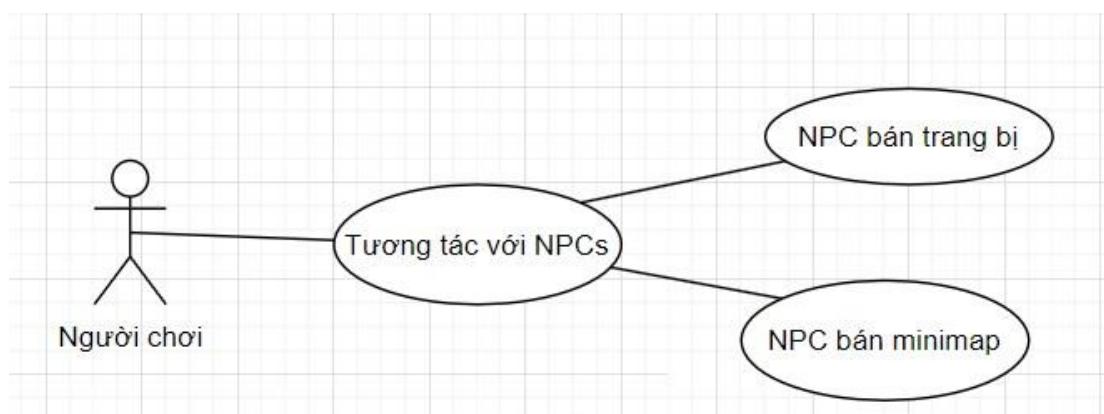
Hình 2.5 Biểu đồ phân rã usecase chức năng giao tranh với quái vật

e) Chức năng thu thập vật phẩm



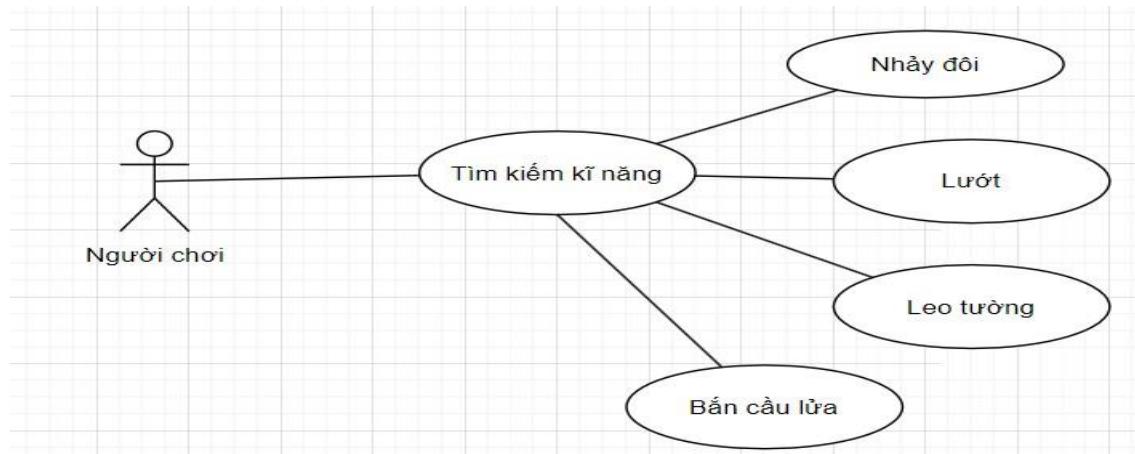
Hình 2.6 Biểu đồ phân rã usecase chức năng thu thập vật phẩm

f) Chức năng tương tác với NPCs



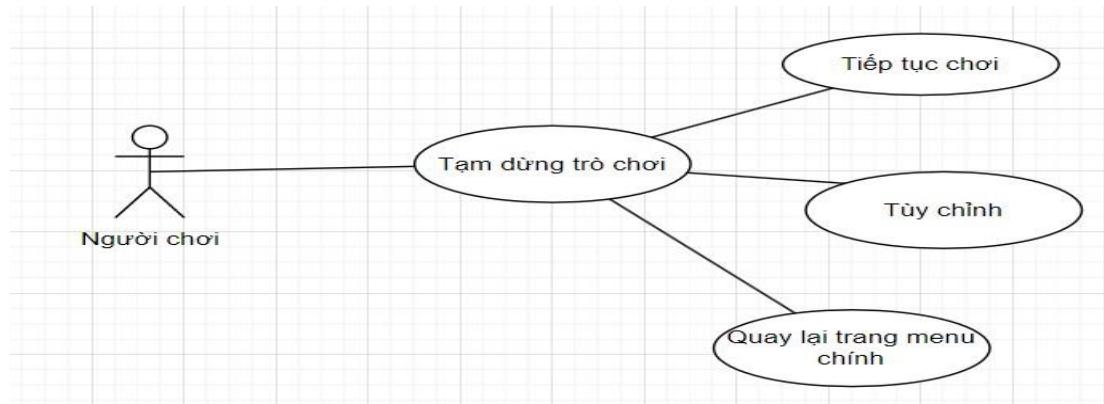
Hình 2.7 Biểu đồ phân rã usecase chức năng tương tác với NPCs

g) Chức năng tìm kiếm kỹ năng



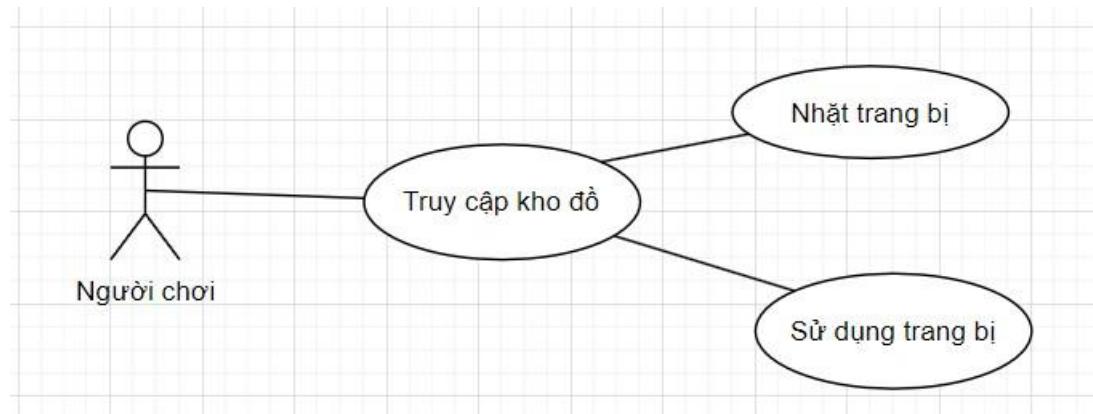
Hình 2.8 Biểu đồ phân rã usecase chức năng tìm kiếm kỹ năng

h) Chức năng tạm dừng trò chơi



Hình 2.9 Biểu đồ phân rã usecase chức năng tạm dừng trò chơi

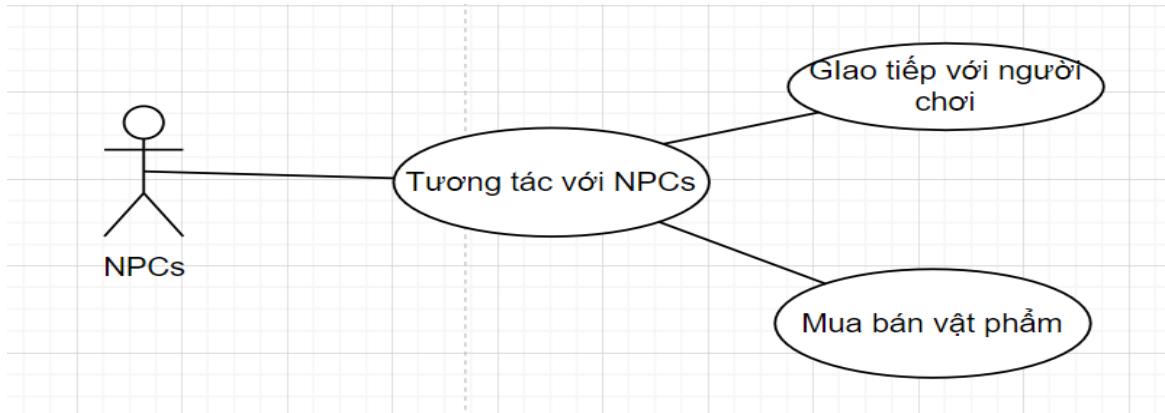
i) Chức năng truy cập kho đồ



Hình 2.10 Biểu đồ phân rã usecase chức năng truy cập kho đồ

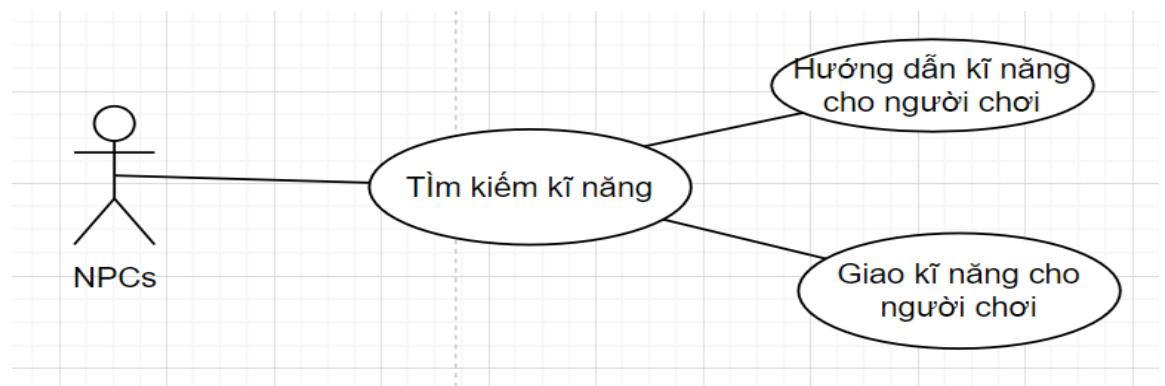
2.1.4.2 Biểu đồ phân rã usecase của tác nhân “NPCs”

a) Chức năng tương tác với NPCs



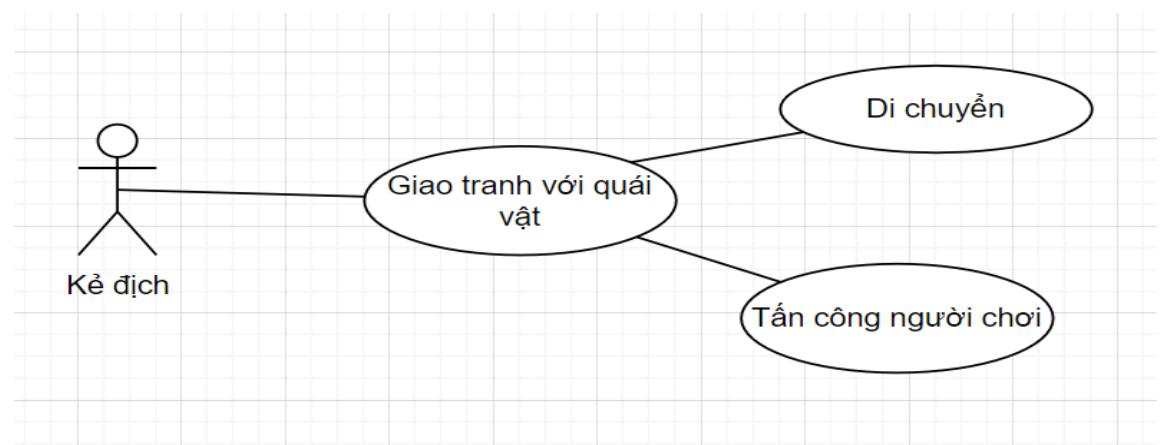
Hình 2.11 Biểu đồ phân rã usecase chức năng tương tác với NPCs của NPCs

b) Chức năng tìm kiếm kĩ năng



Hình 2.12 Biểu đồ phân rã usecase chức năng tìm kiếm kĩ năng của NPCs

2.1.4.3 Biểu đồ phân rã usecase của tác nhân “Kẻ địch”



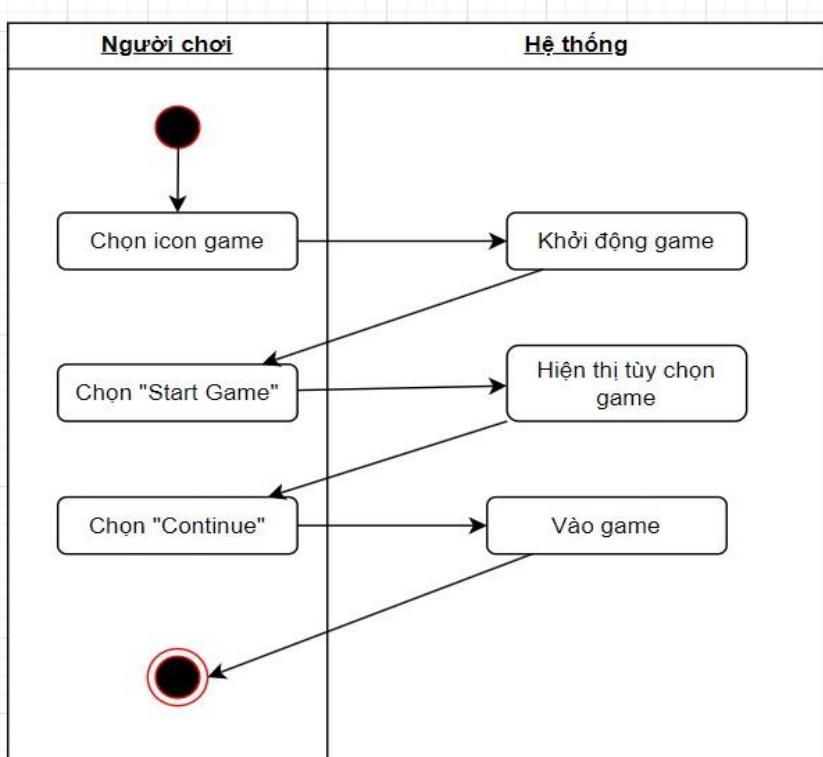
Hình 2.13 Biểu đồ phân rã usecase chức năng giao tranh với quái vật của kẻ địch

2.1.5 Xây dựng kịch bản

2.1.5.1 Kịch bản khởi động game

Bảng 2.2 Bảng kịch bản khởi động game

Tên usecase	Khởi động game
Tác nhân	Người chơi
Tiền điều kiện	Người chơi đã có game trên máy tính
Nội dung	Cho phép người chơi vào game
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người chơi chọn icon game trên máy tính 2. Hiển thị cửa sổ menu game 3. Người chơi chọn “Start Game” 4. Hệ thống chuyển sang cửa sổ chọn chơi game 5. Người chơi chọn “Continue” hoặc “New Game” 6. Hệ thống load dữ liệu game đã lưu lên từ file save 7. Hệ thống chuyển sang cửa sổ vào game
Luồng sự kiện ngoại lệ	<ol style="list-style-type: none"> 1. Người chơi chưa có game trên máy tính 2. File cài đặt bị lỗi khiến game không khởi động được.

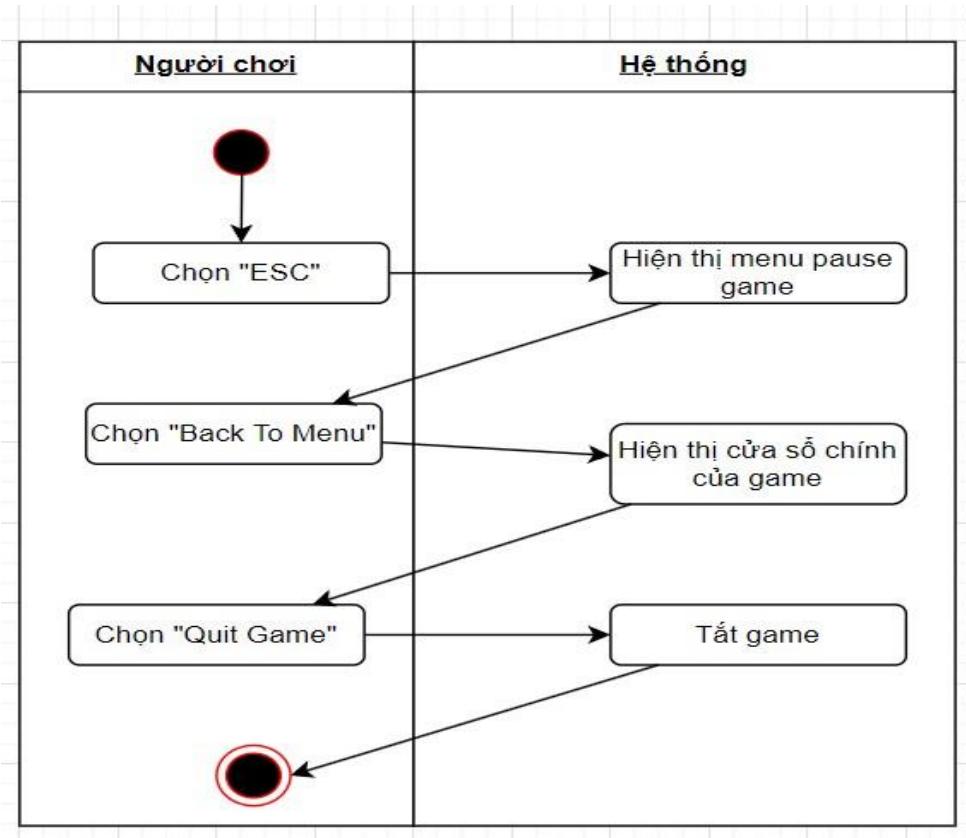


Hình 2.14 Biểu đồ hoạt động chức năng khởi động game

2.1.5.2 Kịch bản thoát game

Bảng 2.3 Bảng kịch bản thoát game

Tên usecase	Thoát game
Tác nhân	Người chơi
Tiền điều kiện	Người chơi đang chơi game trên máy tính
Nội dung	Cho phép người chơi thoát game
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người chơi chọn “ESC” 2. Hiển thị menu pause game 3. Người chơi chọn “Back To Menu” 4. Hệ thống lưu tiến trình chơi và chuyển sang cửa sổ menu game 5. Người chơi chọn “Quit Game” 6. Hệ thống tắt phần mềm game.
Luồng sự kiện ngoại lệ	Không có

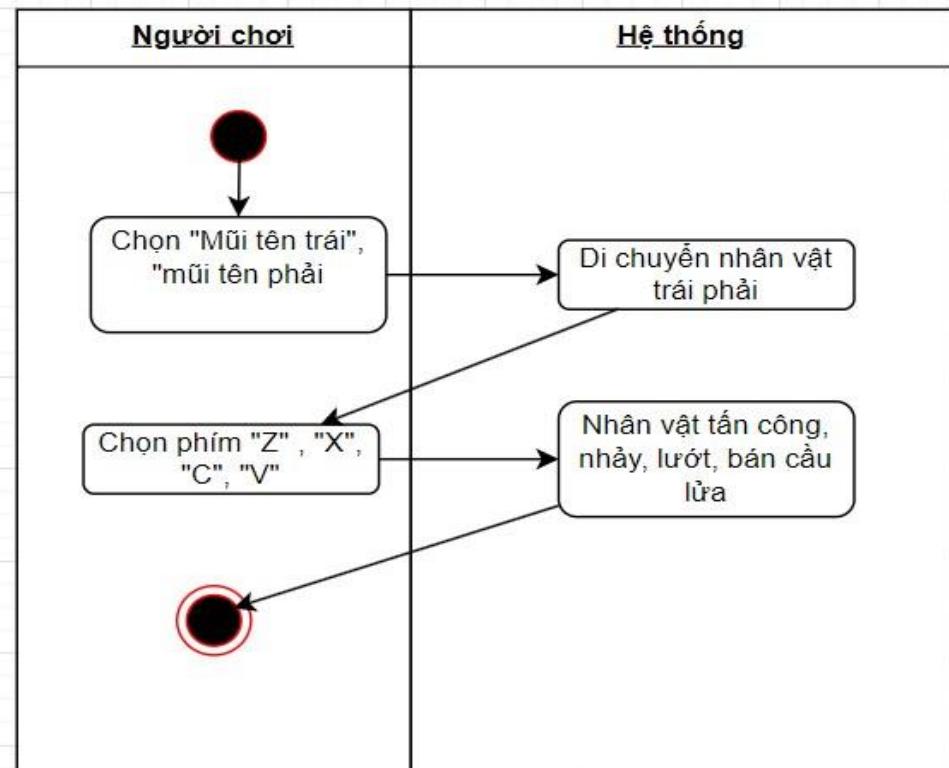


Hình 2.15 Biểu đồ hoạt động chức năng thoát game

2.1.5.3 Kịch bản điều khiển nhân vật

Bảng 2.4 Bảng kịch bản điều khiển nhân vật

Tên usecase	Điều khiển nhân vật
Tác nhân	Người chơi
Tiền điều kiện	Người chơi đang chơi game trên máy tính
Nội dung	Cho phép người chơi điều khiển nhân vật
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người chơi ấn các nút “mũi tên trái” và “mũi tên phải” để di chuyển nhân vật 2. Người chơi ấn các nút “Z” để tấn công 3. Người chơi ấn các nút “X” để nhảy. 4. Người chơi ấn các nút “C” để lướt. 5. Người chơi ấn các nút “V” để bắn cầu lửa.
Luồng sự kiện ngoại lệ	Người chơi chưa nhận được các kỹ năng trong quá trình chơi

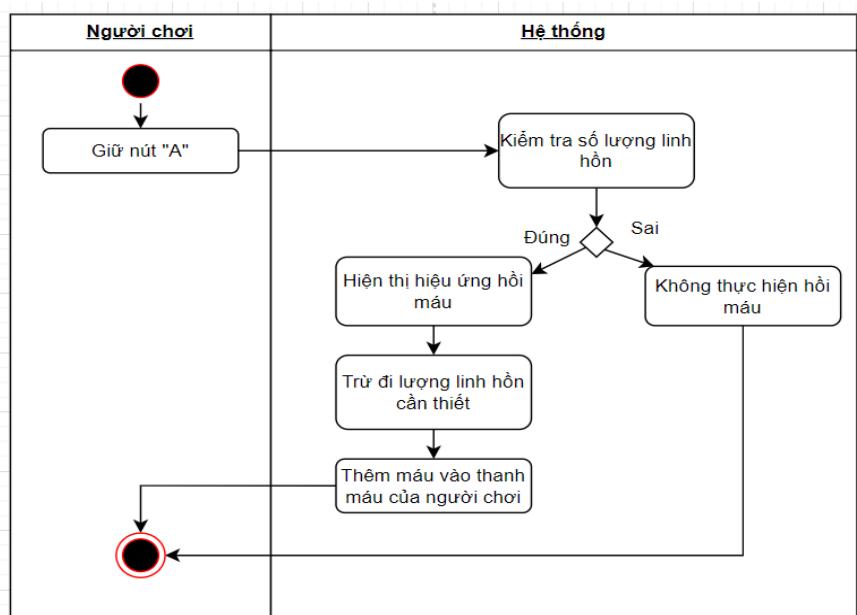


Hình 2.16 Biểu đồ hoạt động chức năng điều khiển nhân vật

2.1.5.4 Kịch bản sử dụng linh hồn để hồi máu

Bảng 2.5 Bảng kịch bản sử dụng linh hồn để hồi máu

Tên usecase	Sử dụng linh hồn để hồi máu
Tác nhân	Người chơi
Tiền điều kiện	1. Người chơi đang chơi game trên máy tính 2. Người chơi có đủ linh hồn để thực hiện hành động hồi máu. 3. Nhân vật của người chơi đang bị mất máu.
Nội dung	Cho phép người chơi sử dụng linh hồn để hồi máu
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người chơi giữ nút “A”. 2. Hệ thống kiểm tra số lượng linh hồn của người chơi. <ol style="list-style-type: none"> a. Nếu đủ linh hồn, chuyển đến bước 3. b. Nếu không đủ linh hồn không thực hiện chức năng. 3. Hệ thống hiển thị hình ảnh và âm thanh của quá trình hồi máu (hiệu ứng đặc biệt). 4. Hệ thống trừ đi một lượng linh hồn cần thiết từ linh hồn của người chơi. 5. Hệ thống thêm một lượng máu tương ứng vào thanh máu của người chơi. 6. Người chơi tiếp tục giữ nút “A”. Quay lại bước 2. 7. Người chơi thả nút "A" để kết thúc quá trình hồi máu.
Luồng sự kiện ngoại lệ	<ol style="list-style-type: none"> 1. Người chơi không có đủ linh hồn để thực hiện hành động hồi máu. 2. Người chơi còn đầy đủ máu 3. Người chơi di chuyển hoặc bị tấn công trong quá trình hồi máu

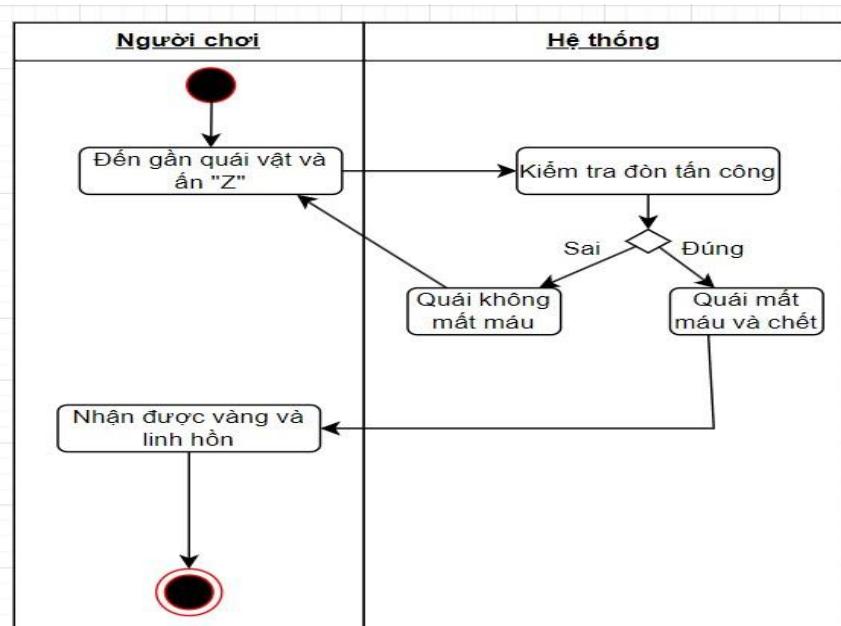


Hình 2.17 Biểu đồ hoạt động chức sử dụng linh hồn để hồi máu

2.1.5.5 Kịch bản giao tranh với quái vật

Bảng 2.6 Bảng kịch bản giao tranh với quái vật

Tên usecase	Giao tranh với quái vật
Tác nhân	Người chơi
Tiền điều kiện	Người chơi đang chơi game trên máy tính
Nội dung	Cho phép người chơi giao tranh với quái vật
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người chơi di chuyển đến gần quái vật. 2. Người chơi ấn nút “Z” để tấn công quái vật. 3. Hệ thống kiểm tra đòn tấn công của người chơi: <ol style="list-style-type: none"> a. Nếu đòn tấn công trúng, hệ thống giảm máu của quái vật. b. Nếu đòn tấn công không trúng, quái vật vẫn còn nguyên máu. 4. Quái vật phản công: <ol style="list-style-type: none"> a. Nếu quái vật tấn công trúng, hệ thống giảm máu của người chơi. b. Nếu quái vật tấn công không trúng, người chơi không bị mất máu. 5. Người chơi tiếp tục tấn công quái vật cho đến khi một trong hai bên bị tiêu diệt. 6. Nếu người chơi tiêu diệt quái vật: Người chơi nhận được vàng và linh hồn. 7. Nếu người chơi bị quái vật tiêu diệt: Người chơi sẽ chết, rơi ra toàn bộ vàng và hồi sinh ở điểm checkpoint.
Luồng sự kiện ngoại lệ	Không có

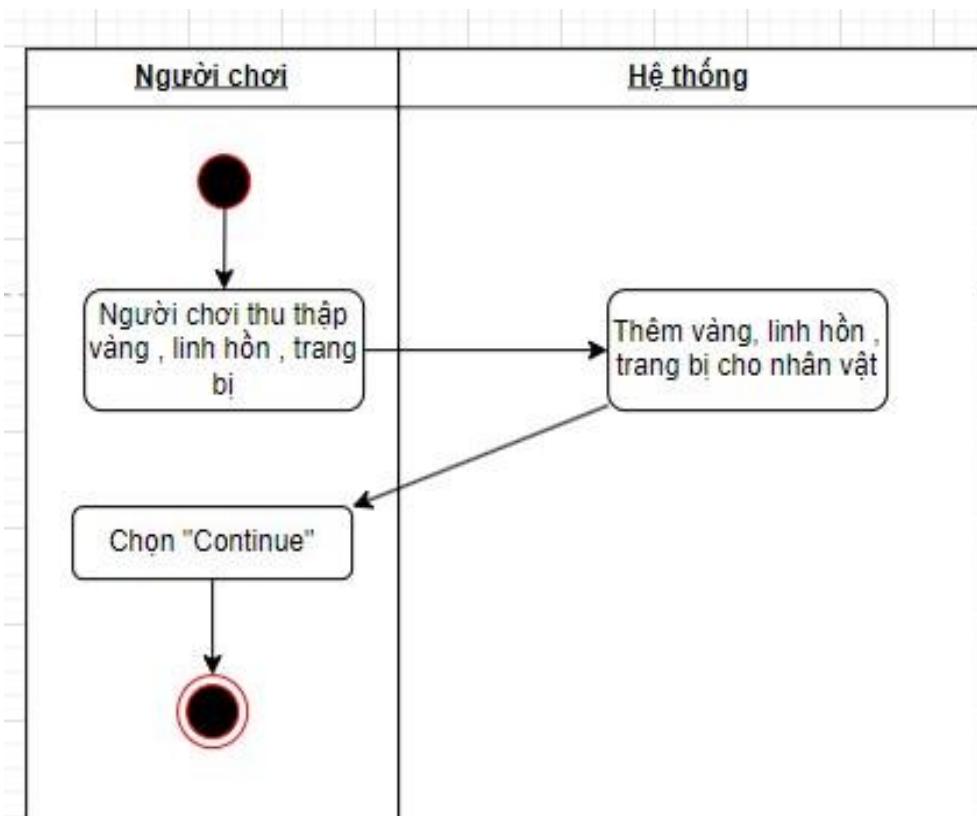


Hình 2.18 Biểu đồ hoạt động chức năng giao tranh với quái vật

2.1.5.6 Kịch bản thu thập vật phẩm

Bảng 2.7 Bảng kịch bản thu thập vật phẩm

Tên usecase	Thu thập vật phẩm
Tác nhân	Người chơi
Tiền điều kiện	Người chơi đang chơi game trên máy tính
Nội dung	Cho phép người chơi thu thập các vật phẩm
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người chơi tấn công quái vật nhận được linh hồn. 2. Người chơi tiêu diệt quái vật nhận được vàng. 3. Người chơi tấn công mỏ quặng nhận được vàng. 4. Người chơi nhặt được trang bị và trang bị sẽ được hệ thống thêm vào kho đồ
Luồng sự kiện ngoại lệ	Không có

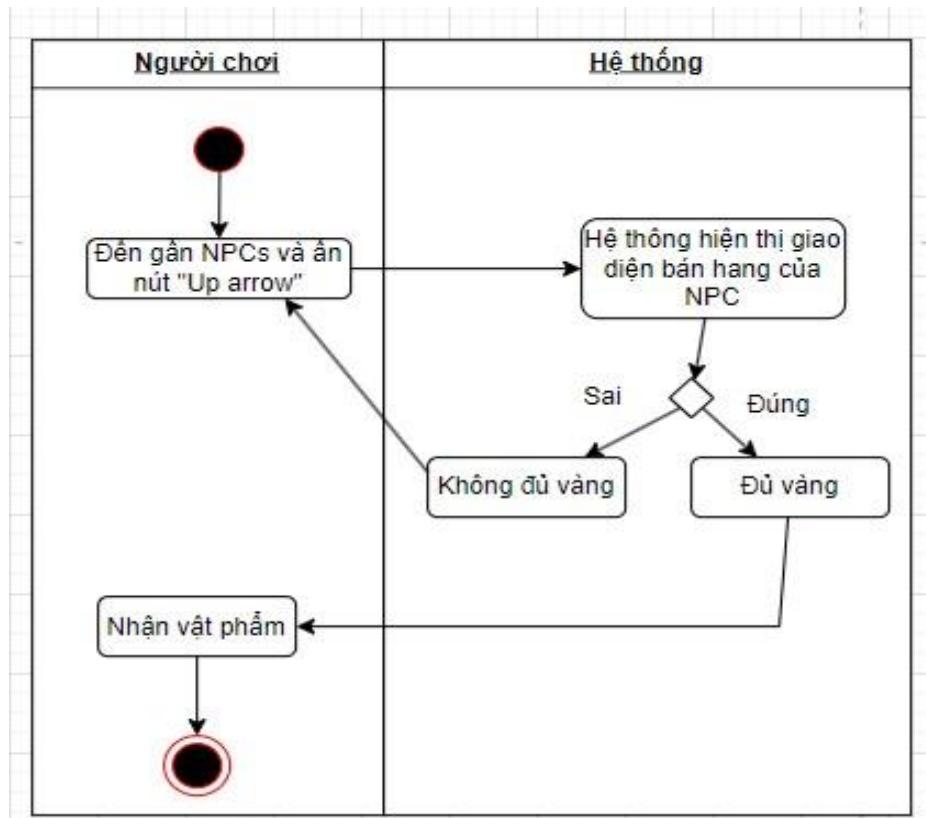


Hình 2.19 Biểu đồ hoạt động chức năng thu thập vật phẩm

2.1.5.7 Kịch bản tương tác với NPCs

Bảng 2.8 Bảng kịch bản tương tác với NPCs

Tên usecase	Tương tác với NPCs
Tác nhân	Người chơi
Tiền điều kiện	Người chơi đang chơi game trên máy tính
Nội dung	Cho phép người chơi tương tác với NPCs
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người chơi di chuyển đến gần NPC. 2. Người chơi ấn nút “Up arrow” để giao tiếp NPC. 3. Hệ thống mở giao diện của NPC bán trang bị. <ol style="list-style-type: none"> a. Người chơi mua trang bị hệ thống sẽ trừ vàng b. Người chơi không mua trang bị. 4. Hệ thống mở giao diện của NPC bán MiniMap: <ol style="list-style-type: none"> a. Người chơi mua MiniMap hệ thống sẽ trừ vàng. b. Người chơi không mua MiniMap.
Luồng sự kiện ngoại lệ	Người chơi chưa vào tầm hoạt động của NPCs.

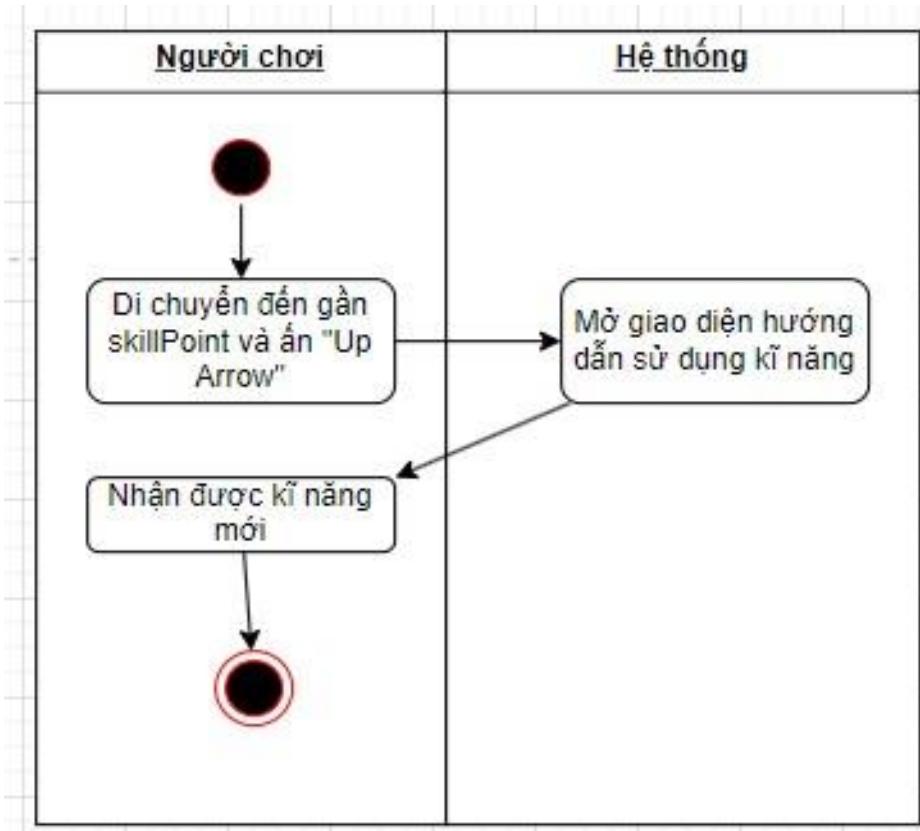


Hình 2.20 Biểu đồ hoạt động chức năng tương tác với NPCs

2.1.5.8 Kịch bản tìm kiếm kỹ năng

Bảng 2.9 Bảng kịch bản tìm kiếm kỹ năng

Tên usecase	Tìm kiếm kỹ năng
Tác nhân	Người chơi
Tiền điều kiện	Người chơi đang chơi game trên máy tính
Nội dung	Cho phép người chơi tìm kiếm kỹ năng
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người chơi di chuyển đến gần Skill Point. 2. Người chơi ấn nút “Up arrow” để nhận kỹ năng. 3. Hệ thống mở giao diện hướng dẫn sử dụng kỹ năng. 4. Người chơi nhận được kỹ năng nhảy đôi, lướt, leo tường, bắn cầu lửa.
Luồng sự kiện ngoại lệ	<ol style="list-style-type: none"> 1. Người chơi chưa vào tầm hoạt động của điểm nhận kỹ năng. 2. Người chơi đã có kỹ năng.

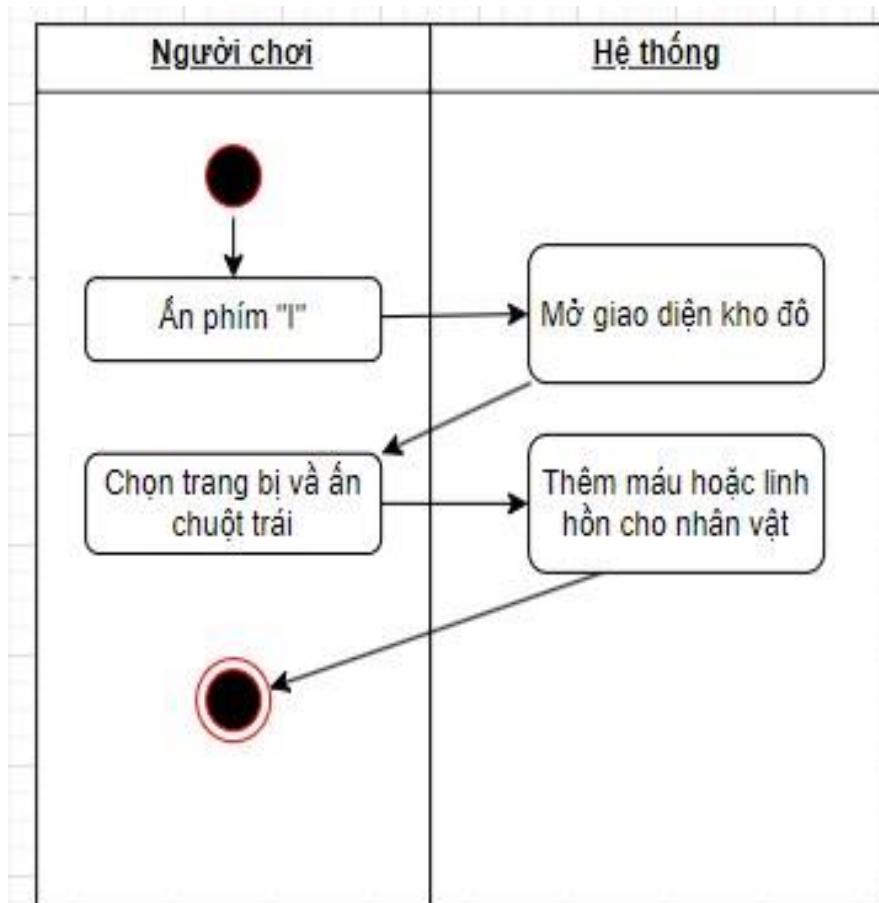


Hình 2.21 Biểu đồ hoạt động chức năng tìm kiếm kỹ năng

2.1.5.9 Kịch bản truy cập kho đồ

Bảng 2.10 Bảng kịch bản truy cập kho đồ

Tên usecase	Truy cập kho đồ
Tác nhân	Người chơi
Tiền điều kiện	Người chơi đang chơi game trên máy tính
Nội dung	Cho phép người chơi truy cập kho đồ
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người chơi ấn phím “I”. 2. Hệ thống mở giao diện kho đồ. 3. Người chơi chọn trang bị và ấn chuột phải để sử dụng. 4. Hệ thống cộng máu tối đa hoặc linh hồn tối đa tùy thuộc và trang bị.
Luồng sự kiện ngoại lệ	<ol style="list-style-type: none"> 1. Người chơi chưa có trang bị.

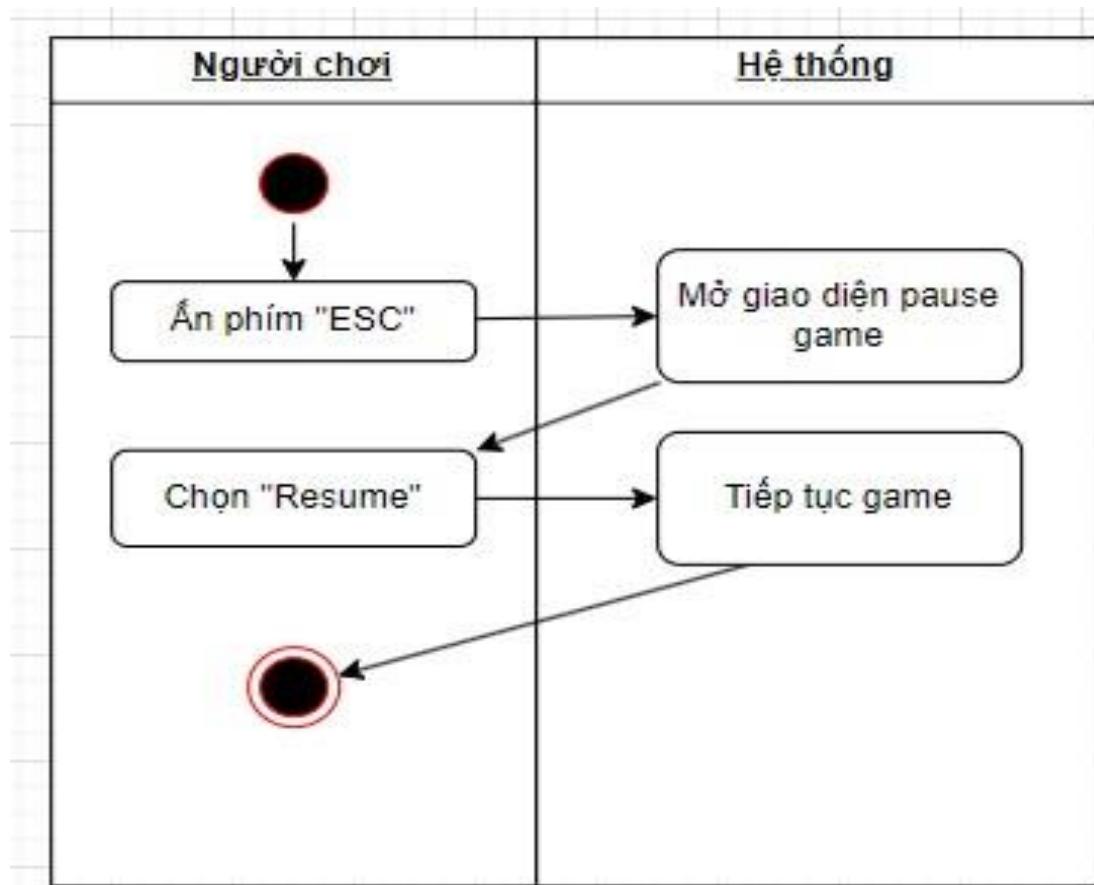


Hình 2.22 Biểu đồ hoạt động chức năng truy cập kho đồ

2.1.5.10 Kích bản tạm dừng trò chơi

Bảng 2.11 Bảng kịch bản tạm dừng trò chơi

Tên usecase	Tạm dừng trò chơi
Tác nhân	Người chơi
Tiền điều kiện	Người chơi đang chơi game trên máy tính
Nội dung	Cho phép người chơi tạm dừng trò chơi
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người chơi ấn phím “ESC”. 2. Hệ thống mở giao diện tạm dừng và dừng trò chơi. 3. Người chơi chọn “Resume” hệ thống tiếp tục trò chơi. 4. Người chơi chọn “Setting” hệ thống hiện giao diện điều chỉnh âm thanh. 5. Người chơi chọn “Back To Menu” hệ thống chuyển cửa sổ menu game.
Luồng sự kiện ngoại lệ	Không có.



Hình 2.23 Biểu đồ hoạt động chức năng tạm dừng trò chơi

2.2 Hệ thống quản lý dữ liệu

Trò chơi sử dụng JSON Serialization và Deserialization để lưu trữ dữ liệu của trò chơi như:

Vàng, linh hồn, vị trí điểm hồi sinh đã mở, các kỹ năng mà người chơi đã mở khóa.

Lưu trữ dữ liệu map cuối cùng người chơi đang sử dụng

Lưu trữ dữ liệu Minimap người chơi đã mua.

Lưu trữ dữ liệu Boss người chơi đã tiêu diệt.

Lưu trữ dữ liệu cửa hàng, mặt hàng đã mua không hiển thị lại nữa.

2.2.1 Lưu trữ và tải lại dữ liệu

Sử dụng JSON Serialization để lưu trữ dữ liệu chuyển đổi dữ liệu của các đối tượng thành dạng chuỗi sau đó lưu xuống file.

```
public virtual void SaveGame()
{
    Debug.Log("SaveGame");
    string jsonString1 = JsonUtility.ToJson(PlayerGold.instance);
    string jsonString2 = JsonUtility.ToJson(SoulManager.instance);
    string jsonString3 = JsonUtility.ToJson(PlayerLife.instance);
    string jsonString4 = JsonUtility.ToJson(PlayerMovement.instance);
    string jsonString5 = JsonUtility.ToJson(PlayerShooting.instance);
    string jsonString6 = JsonUtility.ToJson(MapManager.instance);
    string jsonString7 = JsonUtility.ToJson(MiniMapManager.instance);
    string jsonString8 = JsonUtility.ToJson(BossSave.instance);
    string jsonString9 = JsonUtility.ToJson(ShopSave.instance);

    SaveSystem.SetString(this.GetSaveName1(), jsonString1); // gold
    SaveSystem.SetString(this.GetSaveName2(), jsonString2); // soul, current soul
    SaveSystem.SetString(this.GetSaveName3(), jsonString3); // maxHealth, health, respawnPoint
    SaveSystem.SetString(this.GetSaveName4(), jsonString4); // lockDash, lockDoubleJump, lockSlideWall
    SaveSystem.SetString(this.GetSaveName5(), jsonString5); // lockFireBall
    SaveSystem.SetString(this.GetSaveName6(), jsonString6); // Map Manager
    SaveSystem.SetString(this.GetSaveName7(), jsonString7); // MiniMap Manager
    SaveSystem.SetString(this.GetSaveName8(), jsonString8); // boss save
    SaveSystem.SetString(this.GetSaveName9(), jsonString9); // Shop Manager
}
```

Hình 2.23 Chức năng lưu trữ dữ liệu

Sử dụng JSON Deserialization để lấy dữ liệu từ file lưu trữ sau đó truyền cho các đối tượng trong trò chơi.

```
public virtual void LoadSaveGame()
{
    string jsonString1 = SaveSystem.GetString(this.GetSaveName1());
    string jsonString2 = SaveSystem.GetString(this.GetSaveName2());
    string jsonString3 = SaveSystem.GetString(this.GetSaveName3());
    string jsonString4 = SaveSystem.GetString(this.GetSaveName4());
    string jsonString5 = SaveSystem.GetString(this.GetSaveName5());
    string jsonString6 = SaveSystem.GetString(this.GetSaveName6());
    string jsonString7 = SaveSystem.GetString(this.GetSaveName7());
    string jsonString8 = SaveSystem.GetString(this.GetSaveName8());
    string jsonString9 = SaveSystem.GetString(this.GetSaveName9());

    PlayerGold.instance.FromJson(jsonString1);
    SoulManager.instance.FromJson(jsonString2);
    PlayerLife.instance.FromJson(jsonString3);
    PlayerMovement.instance.FromJson(jsonString4);
    PlayerShooting.instance.FromJson(jsonString5);
    MapManager.instance.FromJson(jsonString6);
    MiniMapManager.instance.FromJson(jsonString7);
    BossSave.instance.FromJson(jsonString8);
    ShopSave.instance.FromJson(jsonString9);
```

Hình 2.24 Chức năng tải lên dữ liệu

2.2.2 Mã hóa dữ liệu

Dữ liệu của trò chơi sẽ được mã hóa và lưu trữ mã hóa XOR (Exclusive OR encryption)

Mã hóa: Đoạn mã này lấy từng ký tự trong chuỗi text, thực hiện phép toán XOR với số 42 và sau đó chuyển đổi lại thành ký tự. Kết quả được thêm vào chuỗi result.

```
static string Crypt(string text)
{
    string result = string.Empty;
    foreach(char j in text) result += (char)((int)j ^ 42);
    return result;
}
```

Hình 2.25 Chức năng mã hóa dữ liệu

Sau đó lưu vào file Profile.bin trong ổ đĩa máy tính của người chơi

This PC > Local Disk (C:) > Users > Admin BVCN88 02 > AppData > LocalLow > DefaultCompany > The Knight Return				
	Name	Date modified	Type	Size
ss	Player.log	7/8/2024 10:50 AM	Text Document	16 KB
ds	Player-prev.log	7/4/2024 2:06 PM	Text Document	23 KB
ste	Profile.bin	7/9/2024 7:47 PM	BIN File	4 KB

Hình 2.26 Vị trí file lưu trữ dữ liệu

CHƯƠNG 3: TỔNG QUAN VỀ GAME “THE KNIGHT RETURN”

3.1 Giới thiệu về game

3.1.1 Thể loại game

Phiêu lưu, hành động, khám phá. Thể loại game 2D.

3.1.2 Yếu tố

Xây dựng nhân vật: Nhẹ nhàng và tinh tế.

Phong cách chiến đấu: Ân tượng và bắt mắt.

Nghệ thuật: Pixel đơn giản nhưng cuốn hút.

Đối tượng: Nhắm tới người chơi trẻ từ 14 tuổi trở lên.

Hệ thống chiến đấu: Đa dạng về quái vật và mức độ, mang đến sự hào hứng và tránh sự nhàm chán. Người chơi sẽ nhận được kỹ năng mới sau mỗi màn chơi.

Phong cách phiêu lưu: Kết hợp với áp lực từ các cạm bẫy và quái vật, lối chơi thử thách tạo nên sự khác biệt và nổi bật hơn so với các game cùng thể loại.

3.1.3 Nội dung

Bối cảnh: Nhân vật chính tĩnh dậy trong một khu rừng bí ẩn. Người chơi điều khiển nhân vật tiến lên phía trước, vượt qua cạm bẫy và quái vật trên hành trình của mình.

Lối chơi: Người chơi sẽ sử dụng kiếm để đánh vào những con quái vật, chúng ẩn nấp, rải rác khắp nơi không để bị tấn công và mất máu.

Thử thách: Lối chơi đòi hỏi sự chú ý và kỹ năng cao của người chơi.

Âm thanh: Mỗi bản đồ có âm thanh riêng, tạo cảm giác căng thẳng và thư giãn khác nhau.

Đồ họa: 2D đẹp mắt kết hợp với nhạc nền xuất sắc là yếu tố quyết định của trò chơi.

3.1.4 Chủ đề

Game nhập vai hành động tiêu diệt quái vật, khám phá thế giới.

3.1.5 Phong cách

Tăm tối, u ám và chết chóc với những con quái vật ghê rợn.

3.1.6 Đối tượng người chơi

Người chơi từ 14 tuổi trở lên, yêu thích nhập vai vào thế giới u ám đầy quái vật đáng sợ, với sự thay đổi liên tục qua các vùng đất khác nhau.

3.1.7 Lối chơi

Menu chính: Người chơi sẽ chọn các mục tương ứng:

Play game: Vào các màn chơi.

Setting: Cài đặt game.

Quit: Thoát khỏi game.

Điều khiển:

Phím mũi tên: Di chuyển nhân vật.

Phím X: Tấn công.

Phím C: Lướt.

Phím Z: Nhảy

Phím ESC: Vào menu Pause, ấn lần nữa để thoát.

3.1.8. Giao diện và cảm nhận

Bối cảnh: Thế giới rộng lớn với những con quái vật ở khắp mọi nơi. Người chơi phải tiến lên phía trước qua các vùng đất để đến được lâu đài.

Phong cách: Tăm tối, kinh dị cả về hình ảnh lẫn âm thanh, tạo cảm giác rùng rợn và kích

thích sự tò mò, ham khám phá của người chơi.

Âm thanh: Nền âm thanh vừa nhẹ nhàng vừa dồn dập, âm thanh từ quái vật và các pha tấn công mạnh mẽ gây kích thích.

3.1.9 Khía cạnh tác động đến người chơi

Hình ảnh và âm thanh: Tác động mạnh mẽ, tạo sự hấp dẫn.

Cách điều khiển và tấn công: Đòi hỏi kỹ năng và sự cẩn trọng.

Cốt truyện: Khiến người chơi tò mò và muốn tìm hiểu thêm.

3.1.10 Mục tiêu trải nghiệm

Nhập vai và cảm xúc: Giúp người chơi hòa nhập vào thế giới trong game, mang lại cảm xúc mà khó có thể trải nghiệm ngoài đời thực. Tạo ra những phút giây thư giãn và giải trí sau những giờ học tập và làm việc căng thẳng.

3.2 Lối chơi và cách vận hành trò chơi

3.2.1. Lối chơi

Trò chơi hành động mang yếu tố chiến lược. Mục tiêu duy nhất là tiêu diệt kẻ thù vượt qua cạm bẫy cho đến khi tìm ra lối thoát, đánh bại boss tìm kiếm các kỹ năng mới và tiến về phía trước. Trò chơi bắt đầu ở một khu rừng tăm tối. Người chơi trải nghiệm thế giới trò chơi qua con mắt của một nhân vật chính. Điều này tạo cảm giác nhập vai cho người chơi, người chơi điều khiển nhân vật chính là một chàng hiệp sĩ bí ẩn, bị mất trí nhớ đi trên hành trình của mình để tìm ra mục tiêu của bản thân, hướng về lâu đài của mình và đánh bại chúa quỷ.

3.2.2 Nhân vật

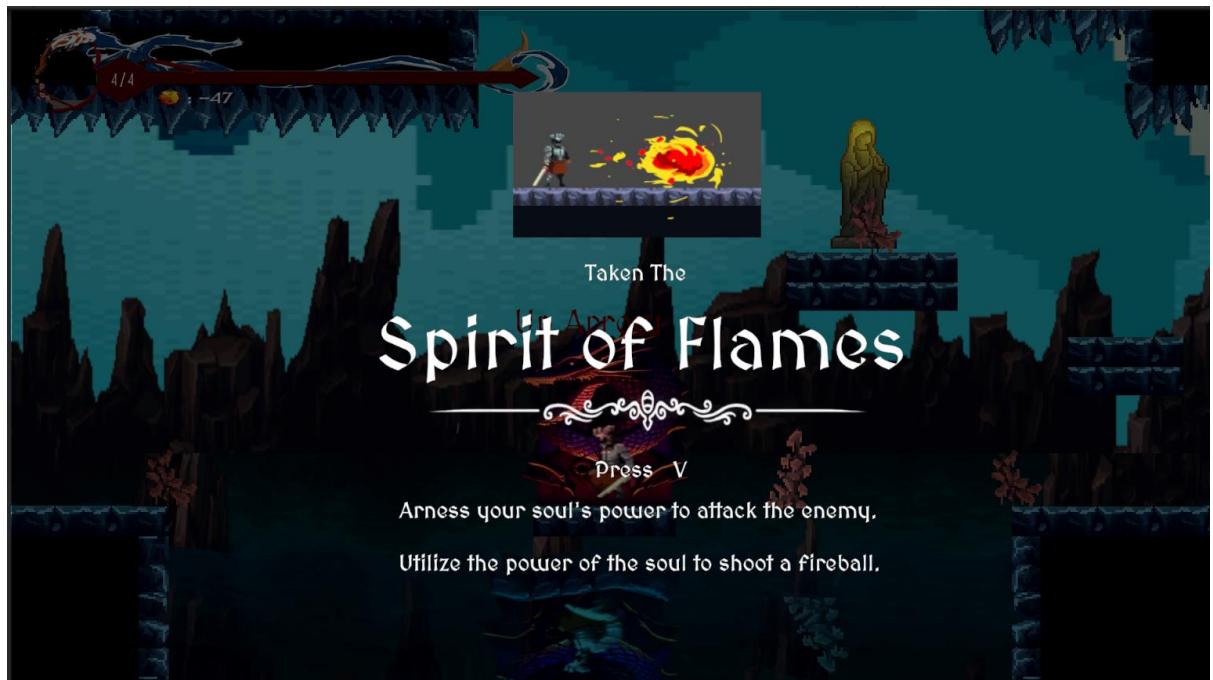
3.2.2.1 Điều khiển nhân vật

Trò chơi có thiết kế chỉ có chế độ chơi đơn người chơi di chuyển tự do trong màn chơi, sử dụng kiếm và các kỹ năng để đánh quái vật. Người chơi có thể di chuyển, nhảy (Gửi nút sẽ nhảy cao, thả luôn sẽ nhảy thấp), nhảy đôi, lướt và leo tường.



Hình 3.1 Điều khiển nhân vật

Người chơi bắt đầu trò chơi với một vũ khí là kiếm, có thể tấn công ngang, tấn công lên trên và tấn công xuống dưới. Khi tấn công xuống dưới vào quái hoặc bẫy sẽ nhảy lên. Người chơi có thể kiếm kĩ năng bắn cầu lửa trong quá trình chơi, sử dụng kĩ năng sê tiêu hao linh hồn.



Hình 3.2 Hướng dẫn sử dụng kỹ năng

3.2.2.2 Máu và linh hồn

Người chơi bắt đầu trò chơi với một số lượng máu và linh hồn tối đa nhất định. Theo mặc định, người chơi có 3 máu, 0 linh hồn, và tối đa lưu trữ được 3 máu, 6 linh hồn. Người chơi có thể tăng trữ lượng tối đa bằng cách sử dụng trang bị.

Khi bị quái vật thì bị giảm 1 máu, khi chạm vào bẫy người chơi sẽ mất máu và hồi sinh ở một địa điểm gần đó. Hiệu quả của sát thương sẽ thay đổi tùy thuộc vào loại quái vật khác nhau. Tuy nhiên, người chơi có thể hồi máu bằng cách tiêu thụ linh hồn, linh hồn kiêm được bằng cách tấn công quái vật.

Người chơi chỉ có một mạng. Khi máu của nhân vật giảm xuống 0, nhân vật chính chết. Khi nhân vật chính chết, người chơi sẽ quay lại điểm hồi sinh gần nhất. Nếu người chơi tiêu diệt tất cả kẻ thù và tiến lên phía trước để qua màn.



Hình 3.3 UI thanh máu, linh hồn, vàng

3.2.2.3 Vàng

Người chơi kiếm được vàng khi tiêu diệt kẻ địch, và tấn công các quặng vàng nằm rải rác quanh bản đồ.

Người chơi sẽ rơi toàn bộ vàng nếu chết. Vàng rơi ra có thể nhặt lại, tuy nhiên nếu người chơi tiếp tục chết trước khi nhặt lại thì vàng sẽ biến mất vĩnh viễn.

3.2.3 Quái vật và bẫy

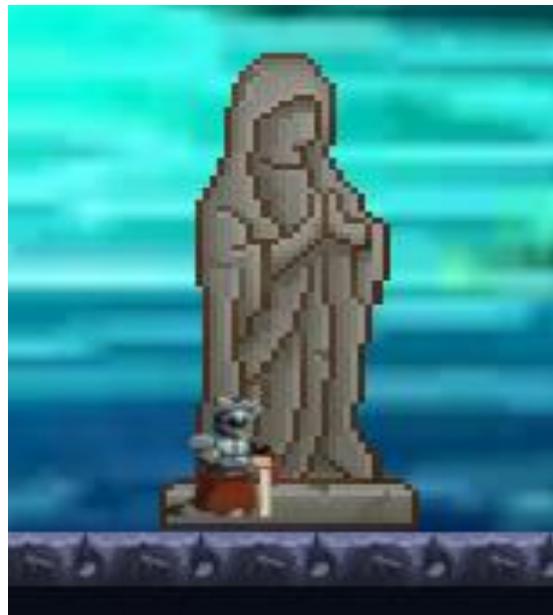
Chúng được sinh ra tại các vị trí trong bản đồ. Số lượng quái vật được giới hạn hữu hạn. Quái tấn công người chơi khi nhân vật đến gần. Mỗi loại quái vật có một mức máu khác nhau. Quái vật bị tấn công, sẽ bị giảm máu. Khi máu của quái vật giảm xuống 0, quái vật sẽ chết rơi ra vàng và biến mất.

Mỗi bản đồ sẽ có những loài quái vật khác nhau và có các kỹ năng riêng biệt tăng tính

thử thách cho người chơi.

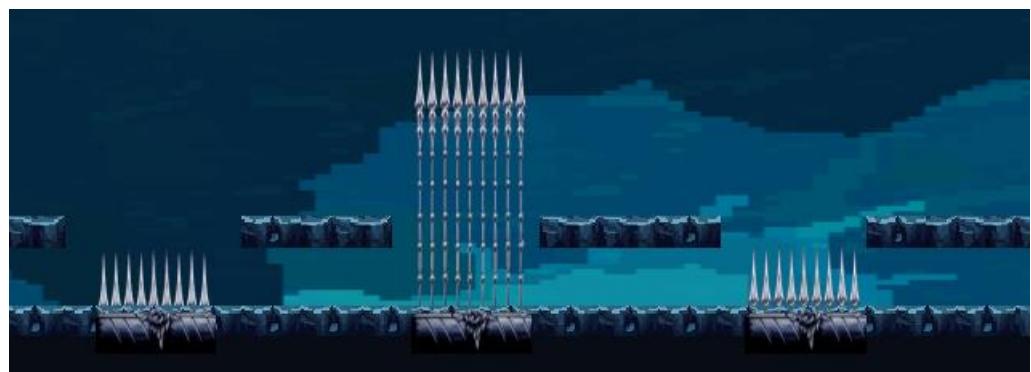
Người chơi có thể tiêu diệt những con boss mạnh mẽ để lấy được một lượng vàng lớn. Nếu người chơi bị boss đánh bại thì người chơi có thể hồi sinh và quay lại phòng boss để tiếp tục tiêu diệt boss. Boss khi bị tiêu diệt sẽ không xuất hiện lại nữa.

Khi người chơi chết và hồi sinh ở điểm hồi sinh thì toàn bộ quái vật sẽ hồi sinh.



Hình 3.4 Điểm hồi sinh

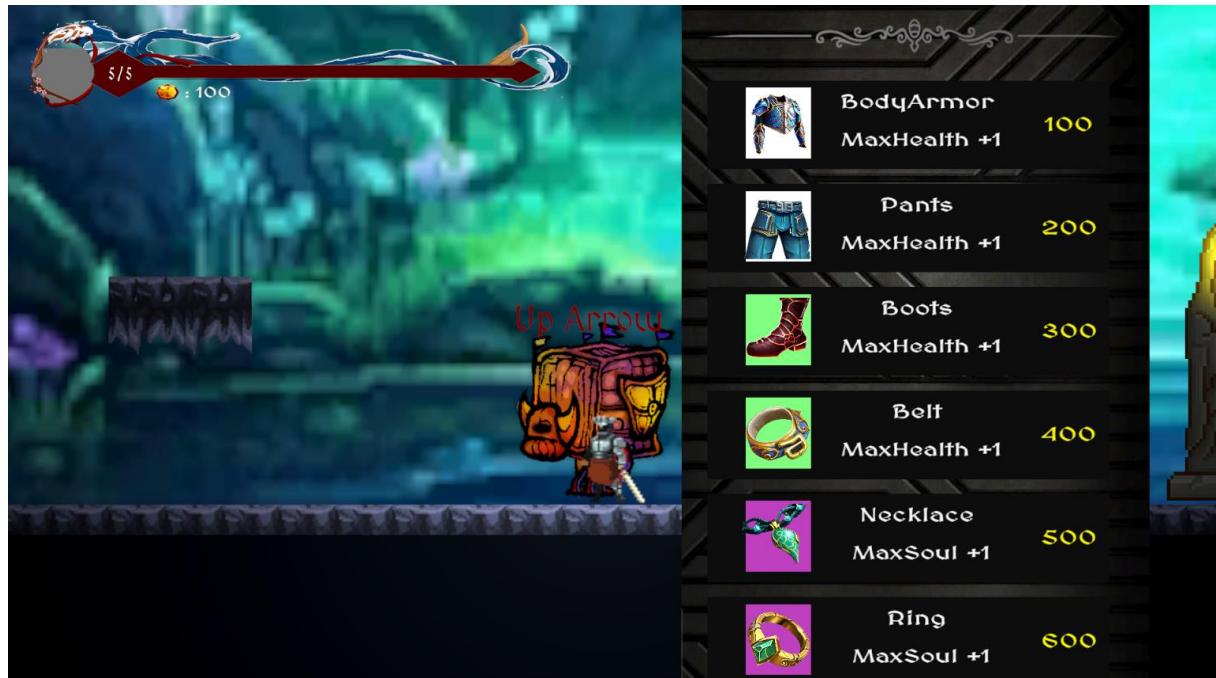
Bẫy rập có ở khắp mọi nơi, nếu người chơi chạm vào bẫy sẽ mất máu và hồi sinh ở một địa điểm gần đó.



Hình 3.5 Cạm bẫy

3.2.4 Cửa hàng và kho đồ

Người chơi có thể sử dụng vàng để mua trang bị trong cửa hàng. Sau khi mua trang bị sẽ rơi ra ở chỗ shopNPC và người chơi có thể nhặt nó.



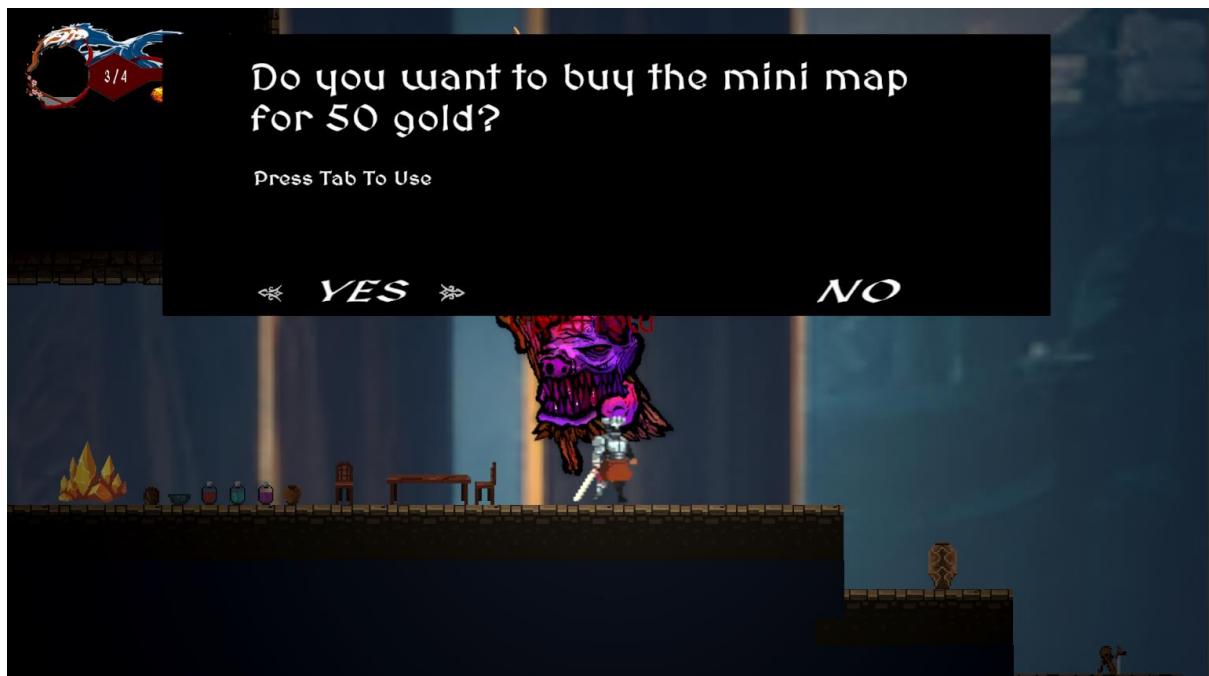
Hình 3.6 Cửa hàng bán trang bị

Người chơi vào kho đồ để kiểm tra và sử dụng trang bị, trang bị sẽ giúp tăng máu và soul tối đa.



Hình 3.7 Kho đồ

Trong quá trình chơi người chơi có thể gặp các Npc bán minimap.



Hình 3.8 Cửa hàng bán minimap

Minimap giúp người chơi xem được toàn bộ bản đồ để tìm kiếm các lối đi.

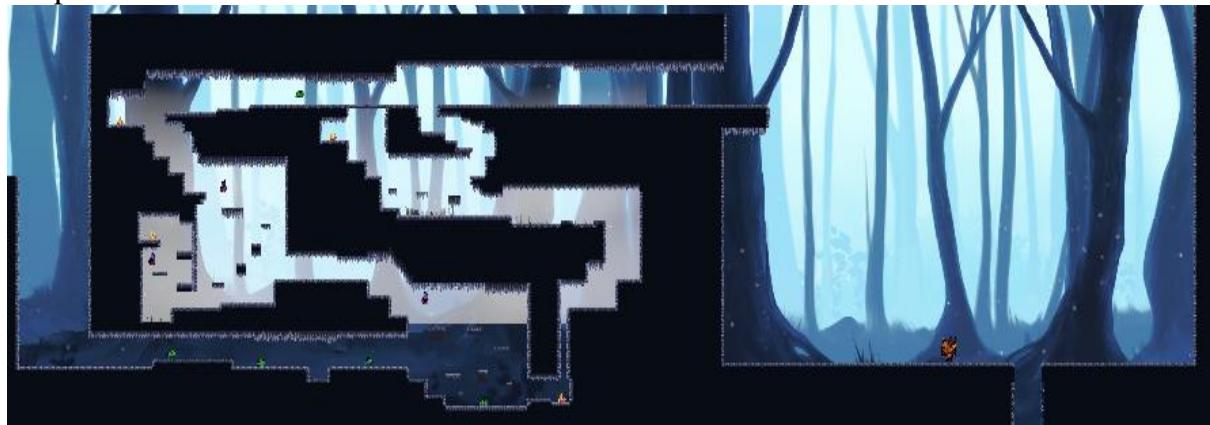


Hình 3.9 Minimap

3.2.5 Level design

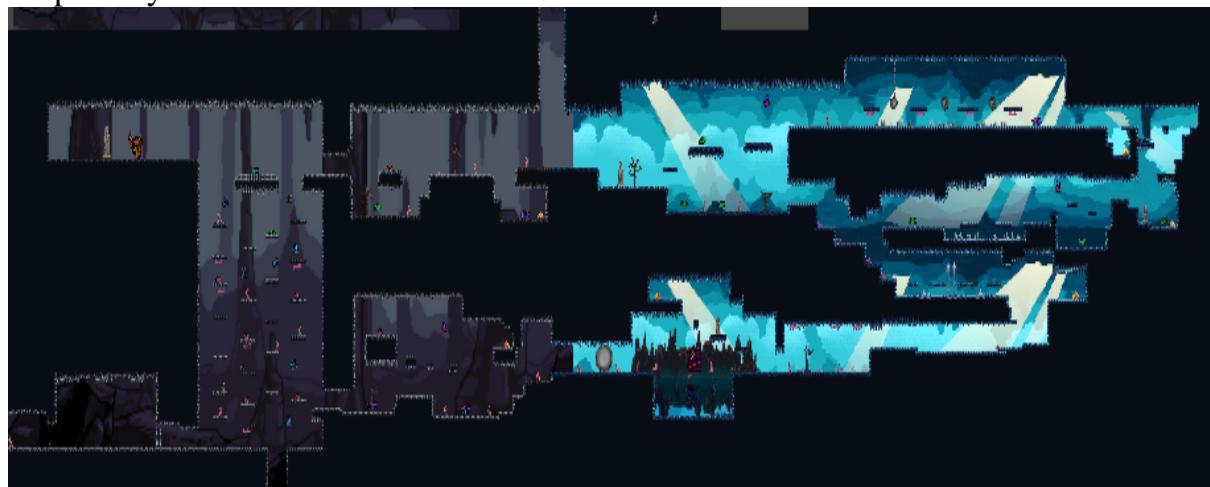
Game sẽ có 5 map khác nhau: Mỗi vùng đất sẽ có những con quái vật và bản nhạc nền khác nhau.

Map 1: Dark Forest



Hình 3.10 Map 1: Dark Forest

Map 2: Mystic Cave



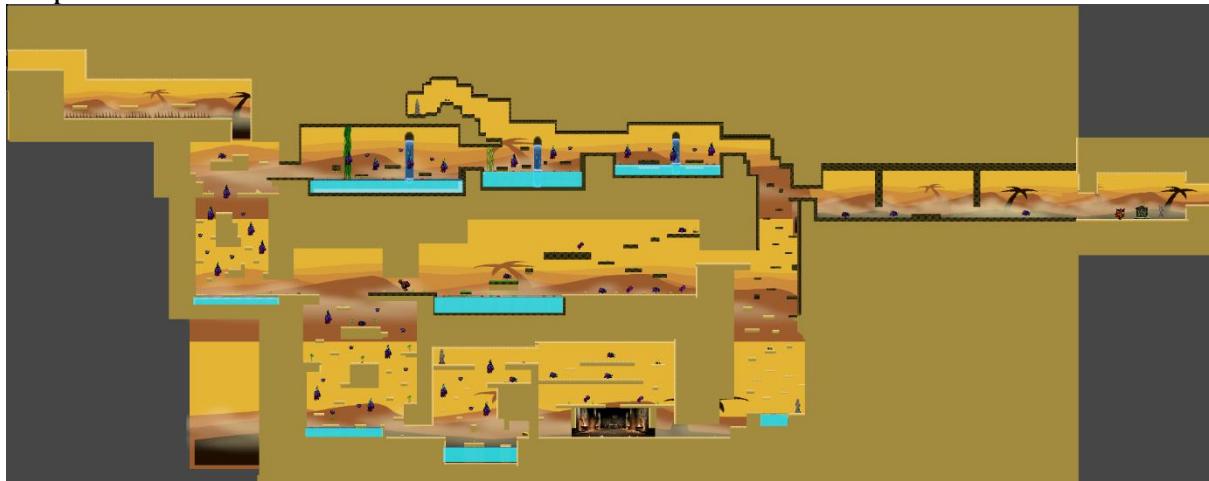
Hình 3.11 Map 2: Mystic Cave

Map 3: Green Forest



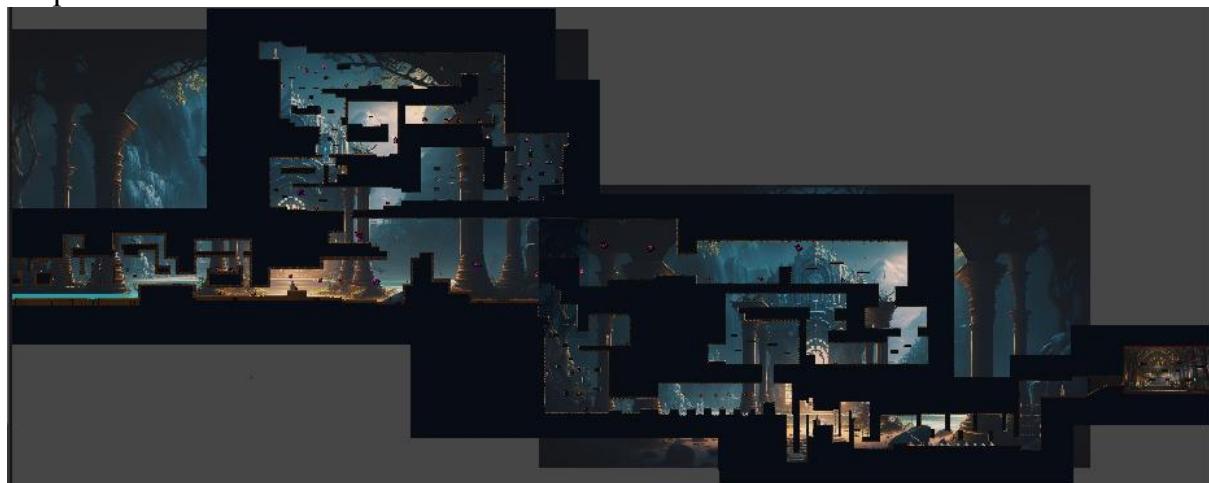
Hình 3.12 Map 3: Green Forest

Map 4: Endless Sands



Hình 3.13 Map 4: Endless Sands

Map 5: Devil's Castle



Hình 3.14 Map 5: Devil's Castle

3.2.6 Game mode

Trò chơi có thiết kế chỉ có chế độ chơi đơn. Trong chế độ chơi đơn, người chơi vượt qua từng map, số lượng kẻ thù và lượng máu bị mất khi bị tấn công sẽ khác nhau. Khi trò chơi ở map cao hơn, số lượng kẻ thù sẽ tăng lên. Bảng dưới đây cung cấp thông tin về số lượng mạnh và kẻ thù trong mỗi cấp độ.

Player (chỉ số giữ nguyên): Khi va chạm với quái sẽ bị -1 máu,

Enemy: Có số lượng máu khác nhau. Tăng dần qua từng map.

3.2.6.1 Quái vật bản đồ Dark Forest, Mystic Cave

Quái vật thường

Bảng 3.1 Bảng quái vật thường map Dark Forest, Mystic Cave

Tên	Máu	Sát thương	Miêu tả
Worm	2 máu	-1 máu	Di chuyển ngang đổi chiều khi gặp tường hoặc vách đá.
Fast Worm	3 máu	-1 máu	Di chuyển ngang đổi chiều khi gặp tường hoặc vách đá. Lao đến khi người chơi đến gần.
Around Worm	2 máu	-1 máu	Di chuyển liên tục vòng quan một tảng đá.
Flying	2 máu	-1 máu	Bay trên cao, lao đến khi người chơi đến gần.
Crazy Flying	2 máu	-1 máu	Bay ngẫu nhiên, lung tung đổi chiều khi gặp tường.



Hình 3.15 Worm



Hình 3.16 Fast Worm



Hình 3.17 Around Worm



Hình 3.18 Flying



Hình 3.19 Crazy Flying

Quái vật Boss:

Bảng 3.2 Quái vật boss map Dark Forest, Mystic Cave

Tên	Máu	Sát thương	Miêu tả kĩ năng
BlueBirdAngry	15 máu	-1 máu	<ul style="list-style-type: none"> - Move: Di chuyển lên xuống tự túc. - Attack Player: Lao nhanh tới người chơi. - Attack Up Down: Lao lên xuống điên cuồng với tốc độ nhanh.
MadHog	25 máu	-1 máu	<ul style="list-style-type: none"> - Jump Attack: Nhảy về phía người chơi. - Dash Attack: Lao nhanh về phía người chơi. - Rampage Attack: Tấn công liên tục sang 2 bên trái phải gây ra động đất làm rơi những tảng đá nhẹ, - WaveAttack: Tạo ra một đợt sóng bay về phía người chơi.



Hình 3.20 BlueBirdAngry



Hình 3.21 MadHog

3.2.6.2. Quái vật bản đồ Green Forest

Quái vật thường

Bảng 3.3 Bảng quái vật thường map Green Forest

Tên	Máu	Sát thương	Miêu tả
Walk Tree	3 máu	-1 máu	Di chuyển ngang đổi chiều khi gặp tường hoặc vách đá.
Jump Frog	3 máu	-1 máu	Di chuyển ngang đổi chiều khi gặp tường hoặc vách đá. Lao đến khi nhân vật đến gần. Nhảy đến nhân vật khi đến gần.
Flying Frog	3 máu	-1 máu	Bay trên cao, lao đến khi nhân vật đến gần.
Boom Worm	3 máu	-1 máu	Mỗi lần bị tấn công sẽ phát nổ gây sát thương cho nhân vật.
Explosion Flower	3 máu	-1 máu	Khi bị tiêu diệt sẽ phát nổ gây sát thương cho nhân vật.
Mom Butterfly	4 máu	-1 máu	Đến gần nhân vật sẽ sinh ra Litter Butterfly.
Litter Butterfly	2 máu	-1 máu	Được sinh ra từ Mom Butterfly khi sinh ra sẽ lao lên nhân vật và gây sát thương.
Platform Worm	Không thể chết	-1 máu	Sinh sống ở các dòng sông. Người chơi có thể nhảy lên, tấn công xuống để nảy lên và vượt qua các dòng sông.



Hình 3.22 Walk Tree



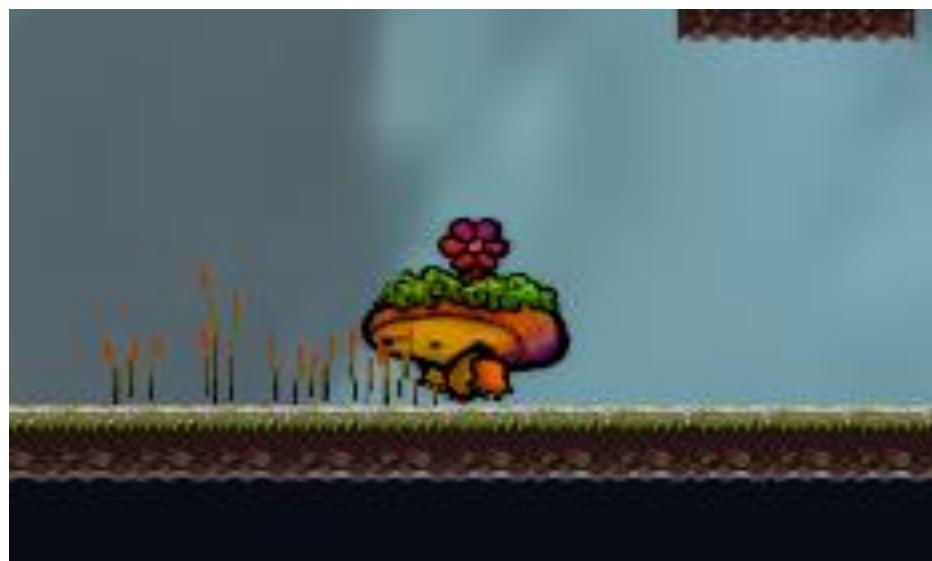
Hình 3.23 Jump Frog



Hình 3.24 Flying Frog



Hình 3.25 Boom Worm



Hình 3.26 Explosion Flower



Hình 3.27 Mom Butterfly & Litter Butterfly



Hình 3.28 Platform Worm

Quái vật Boss

Bảng 3.4 Quái vật boss map Green Forest

Tên	Máu	Sát thương	Miêu tả kĩ năng
Fairy Poison	15 máu	-1 máu	<ul style="list-style-type: none"> - Move: Di chuyển về phía người chơi. - PoisonBall attack: Bắn ra 3 lượt đạn, mỗi lần có 5 viên PoisonBall theo hình nón.
Tusk	30 máu	-1 máu	<ul style="list-style-type: none"> - Jump Attack: Nhảy 3 lần về phía người chơi. - Move Attack: Lao đến nhân vật 2 lần. - PoisonBall Attack: Bắn ra 3 lượt đạn, một lượt có 1 chùm PoisonBall bay về phía nhân vật. - Top PoisonBall Attack: Nhảy lên cao và thả PoisonBall rơi xuống người chơi.



Hình 3.29 Fairy Poison



Hình 3.30 Tusk

3.2.6.3 Quái vật bản đồ Endless Sands

Quái vật thường

Bảng 3.5 Bảng quái vật thường map Endless Sands

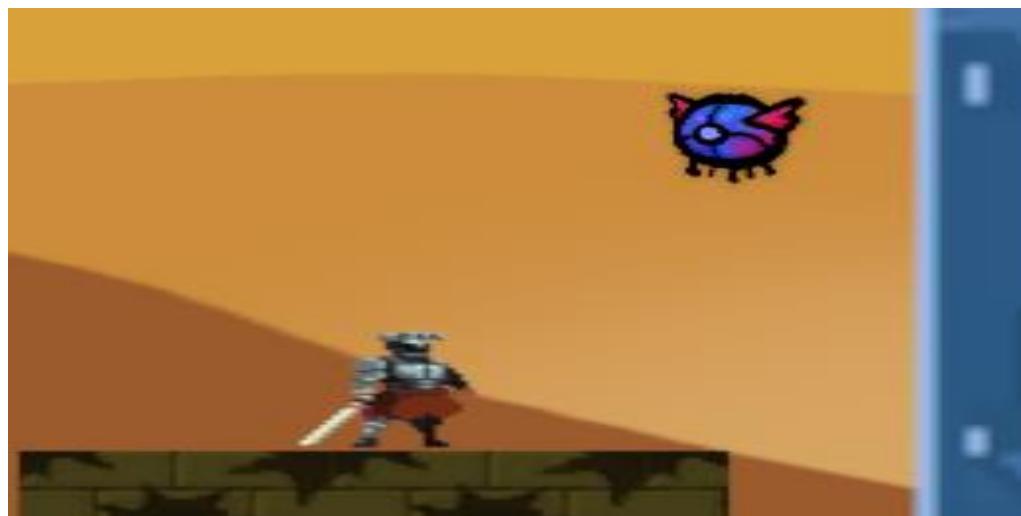
Tên	HP	Sát thương	Miêu tả
DogMachine	3 máu	-1 máu	Di chuyển ngang đổi chiều khi gặp tường hoặc vách đá.
Lamp Machine	3 máu	-1 máu	Di chuyển ngang đổi chiều khi gặp tường hoặc vách đá. Lao đến khi người chơi đến gần. Nhảy đến người chơi khi đến gần.
OneEyeMachine	2 máu	-1 máu	Bay trên cao, không di chuyển đứng im chặn đường người chơi.
BigEyeMachine	3 máu	-1 máu	Bay trên cao, di chuyển lên xuống chặn đường người chơi.



Hình 3.31 DogMachine



Hình 3.32 Lamp Machine



Hình 3.33 OneEyeMachine



Hình 3.34 BigEyeMachine

Quái vật Boss

Bảng 3.6 Quái vật boss map Endless Sands

Tên	Máu	Sát thương	Miêu tả kĩ năng
RaMachine	15 máu	-1 máu	<ul style="list-style-type: none"> - NearAttack: Khi người chơi đến gần sẽ dùng mỏ để tấn công. - FireBall Attack: Bắn ra 1 môt chùm đạn về phía người chơi.
NuclearMachine	30 máu	-1 máu	<ul style="list-style-type: none"> - Move FireBall Attack: Vừa di chuyển vừa bắn những quả cầu lửa về phía người chơi. - JumpAttack: Nhảy về phía người chơi, sau đó nhảy lại về vị trí cũ. - Rampage FireBall Attack: Bắn ra 3 lượt đạn, mỗi một lượt có 1 chùm FireBall bay về phía người chơi.



Hình 3.35 RaMachine



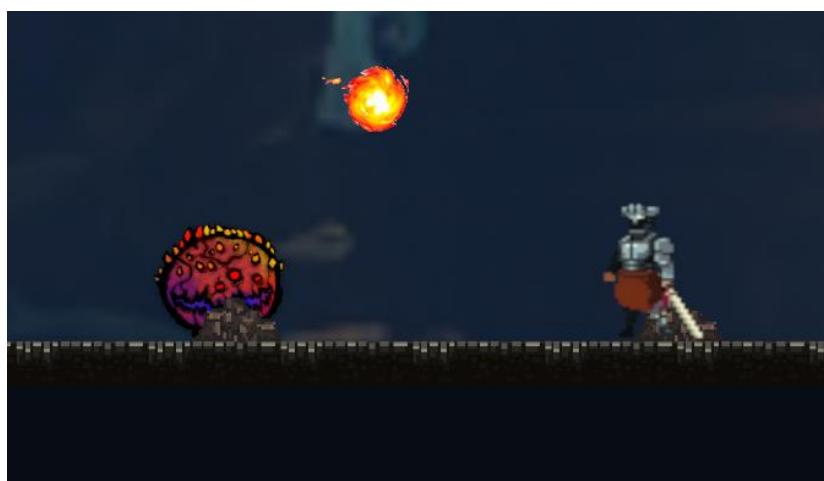
Hình 3.36 NuclearMachine

3.2.6.4 Quái vật bản đồ Devil's Castle

Quái vật thường

Bảng 3.7 Bảng quái vật thường map Devil's Castle

Tên	HP	Sát thương	Miêu tả
FireDevil	3 máu	-1 máu	Di chuyển ngang đổi chiều khi gặp tường hoặc vách đá. Khi nhân vật đến gần bắn 1 quả cầu lửa về phía người chơi.
FlyDevil	3 máu	-1 máu	Bay trên cao, khi nhân vật đến gần bắn 1 quả cầu lửa về phía người chơi.
DevilHog	2 máu	-1 máu	Bay trên cao, khi nhân vật đến gần bắn 3 quả cầu lửa về phía người chơi.
JumpDevil	3 máu	-1 máu	Di chuyển bằng cách nhảy liên tục, khi người chơi đến gần thì lao đến người chơi.
WallWorm	3 máu	-1 máu	Di chuyển dọc ở các mặt tường gây cản trở cho người chơi.



Hình 3.37 FireDevil



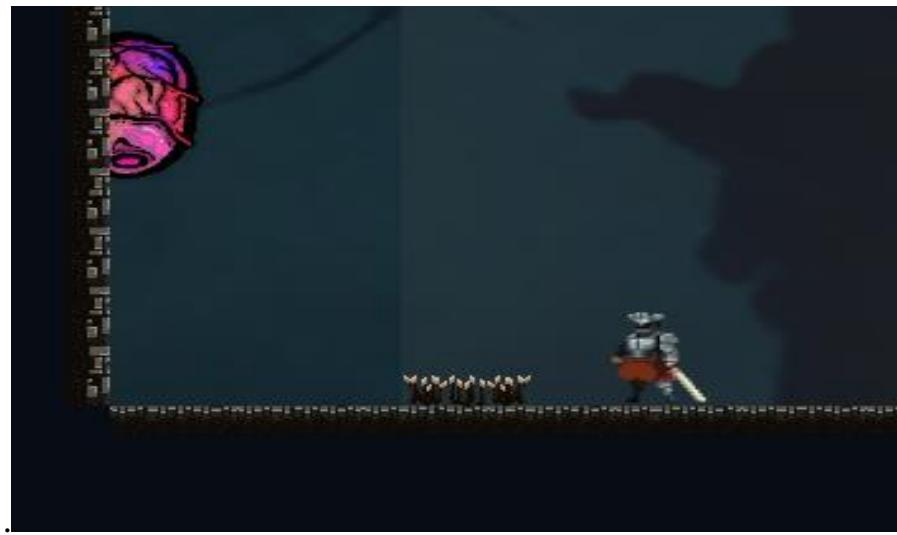
Hình 3.38 FlyDevil



Hình 3.39 DevilHog



Hình 3.40 JumpDevil



Hình 3.41 WallWorm

Quái vật Boss:

Bảng 3.8 Quái vật boss map Devil's Castle

Tên	Máu	Sát thương	Miêu tả kĩ năng
PhoenixMachine	25 máu	-1 máu	<ul style="list-style-type: none"> - Move And SwordAttack: Liên tục di chuyển và tạo ra những thanh kiếm lao về phía người chơi. - Shield move: Đứng im liên tục phóng to và thu nhỏ phạm vi khiên.
Brings Of Death	40 máu	-1 máu	<ul style="list-style-type: none"> - Moving: Di chuyển về phía người chơi. - JumpAttack: Nhảy lên và chém xuống gây sát thương. - WaveAttack: Bắn ra 3 quả cầu lửa hình nón. Có 3 đợt tấn công. - Spawn enemy: Sinh ra những con quái nhỏ bay về phía người chơi.

			<ul style="list-style-type: none"> - CastSpell: Triệu hồi cơn bão lao về phía người chơi gây sát thương liên tục. - Meteor attack: Triệu hồi hàng loạt thiên thạch bay ngẫu nhiên.
--	--	--	--



Hình 3.42 PhoenixMachine



Hình 3.43 Brings Of Death

3.3. Game Control

Trò chơi sẽ sử dụng chuột và bàn phím để điều khiển

Movement

- Left – Left Arrow
- Right – Right Arrow
- Jump – X
- Dash – C

Actions

- Attack Weapon - Z
- Shooting FireBall – V
- Healing – Hold A
- Open MiniMap - Hold Tab
- Look Up – Hold Up Arrow
- Look Down – Hold Down Arrow
- Open Inventory – I
- Menu – ESC

3.4. Thắng và thua

Mục tiêu của trò chơi là tiêu diệt tất cả kẻ thù trước khi chúng giết người chơi.

Người chơi chỉ có 1 mạng với số lượng máu nhất định. Khi máu của player giảm xuống 0, người chơi sẽ chết => Sẽ dịch chuyển đến điểm hồi sinh gần nhất (Tượng Thần)

Người chơi cần phải đánh bại boss cuối để xuất hiện cánh cổng dẫn tới kết thúc của trò chơi.



Hình 3.44 Cổng kết thúc game

CHƯƠNG 4: CÀI ĐẶT TRÒ CHƠI VÀ KẾT LUẬN

4.1 Môi trường cài đặt

Để phát triển trò chơi The Knight Return, em đã sử dụng máy tính của mình là , Intel(R) Core(TM) i7-7820HQ CPU @ 2.90GHz 2.90 GHz, ổ cứng SSD 250GB chạy hệ điều hành Windows 10 và cài thêm các môi trường phát triển sau:

4.1.1 Unity Hub

Unity Hub là một công cụ quản lý mạnh mẽ và thiết yếu trong hệ sinh thái Unity, được thiết kế để giúp các nhà phát triển dễ dàng quản lý nhiều phiên bản Unity Editor và các dự án Unity của họ. Unity Hub mang lại nhiều lợi ích và tính năng quan trọng như sau:

4.1.1.1 Cài đặt và Quản lý Phiên bản Unity

Unity Hub cho phép bạn tải xuống và cài đặt nhiều phiên bản Unity Editor khác nhau, từ các phiên bản mới nhất đến các phiên bản cũ hơn. Điều này rất hữu ích khi bạn cần làm việc với các dự án yêu cầu các phiên bản Unity khác nhau hoặc muốn thử nghiệm các tính năng mới của phiên bản beta.

Bạn có thể dễ dàng chuyển đổi giữa các phiên bản Unity đã cài đặt mà không cần phải gỡ bỏ hoặc cài đặt lại bất kỳ phiên bản nào, giúp tiết kiệm thời gian và công sức.

4.1.1.2 Quản lý Dự án

Unity Hub cung cấp một giao diện người dùng trực quan để quản lý tất cả các dự án Unity của bạn. Bạn có thể tạo mới, mở, sao chép hoặc xóa các dự án từ một vị trí tập trung.

Khi tạo một dự án mới, Unity Hub cho phép bạn chọn phiên bản Unity sẽ sử dụng, thiết lập đường dẫn lưu trữ dự án và cài đặt các module bổ sung như hỗ trợ cho nền tảng di động, VR/AR, và các dịch vụ mạng.

4.1.1.3 Tài nguyên Học tập và Mẫu Dự án

Unity Hub tích hợp các tài nguyên học tập phong phú từ Unity Learn, bao gồm các khóa

học, hướng dẫn, và video giúp bạn học cách sử dụng Unity một cách hiệu quả.

Ngoài ra, Unity Hub cũng cung cấp quyền truy cập nhanh chóng đến các mẫu dự án (template) được thiết kế sẵn cho các thể loại game và ứng dụng khác nhau. Những mẫu này có thể giúp bạn bắt đầu dự án nhanh hơn và học hỏi từ các thiết lập và cấu trúc đã có.

4.1.1.4 Tài khoản và Dịch vụ Unity

Unity Hub tích hợp với tài khoản Unity, cho phép bạn đăng nhập và đồng bộ hóa với các dịch vụ trực tuyến của Unity như Unity Teams, Unity Collaborate, Asset Store, Unity Analytics, và Unity Cloud Build.

Điều này giúp bạn quản lý các giấy phép sử dụng, theo dõi tiến độ dự án, và hợp tác với các thành viên trong nhóm một cách hiệu quả hơn.

4.1.2 Unity Editor

Unity Editor là môi trường phát triển tích hợp (IDE) của Unity, cung cấp một bộ công cụ toàn diện để xây dựng và phát triển các trò chơi, ứng dụng tương tác và trải nghiệm thực tế ảo. Unity Editor bao gồm nhiều tính năng mạnh mẽ và đa dạng:

4.1.2.1 Giao diện Người dùng

Unity Editor có giao diện đồ họa trực quan, được chia thành nhiều cửa sổ và bảng điều khiển khác nhau như Scene, Game, Hierarchy, Project, Inspector, Console, và Asset Store. Mỗi cửa sổ có một chức năng cụ thể và có thể được tùy chỉnh theo nhu cầu của người dùng.

Scene view cho phép bạn thiết kế và bố trí các thành phần trong game của mình, bao gồm các đối tượng 3D, ánh sáng, camera, và nhiều yếu tố khác. Bạn có thể di chuyển, xoay, và thay đổi kích thước các đối tượng một cách dễ dàng bằng các công cụ trực quan.

4.1.2.2 Script và Công cụ Lập trình

Unity sử dụng C# làm ngôn ngữ lập trình chính. Các script được viết bằng C# giúp điều khiển hành vi của các đối tượng trong game, xử lý logic trò chơi, và tương tác với người dùng.

Unity Editor tích hợp với các công cụ lập trình như Visual Studio và Visual Studio Code, cung cấp một môi trường lập trình mạnh mẽ với các tính năng như tự động hoàn thành mã, gỡ lỗi (debugging), và quản lý dự án.

4.1.2.3 Hệ thống Vật lý và Animation

Unity Editor cung cấp các công cụ mạnh mẽ để tạo ra các hiệu ứng vật lý và animation phức tạp. Hệ thống vật lý của Unity cho phép bạn mô phỏng các tương tác vật lý như va chạm, lực hấp dẫn, và động lực học.

Hệ thống animation của Unity cho phép bạn tạo và quản lý các animation cho các đối tượng 3D, từ các chuyển động đơn giản đến các hoạt cảnh phức tạp. Bạn có thể sử dụng Animator và Animation windows để thiết kế và điều chỉnh các animation của mình.

4.1.2.4 Tích hợp Asset Store

Unity Editor tích hợp với Unity Asset Store, nơi bạn có thể mua hoặc tải về các asset để sử dụng trong dự án của mình. Asset Store cung cấp một kho tài nguyên phong phú bao gồm mô hình 3D, texture, âm thanh, script, công cụ, và nhiều tài nguyên khác.

Việc sử dụng các asset từ Asset Store có thể giúp bạn tiết kiệm thời gian và công sức trong quá trình phát triển dự án, cũng như học hỏi từ các tài nguyên chất lượng cao.

4.1.2.5 Khả năng Xuất bản Đa nền tảng

Unity cho phép bạn xuất bản trò chơi và ứng dụng lên nhiều nền tảng khác nhau, bao gồm Windows, macOS, iOS, Android, WebGL, PlayStation, Xbox, Nintendo Switch, và nhiều nền tảng khác. Điều này giúp bạn tiếp cận được với nhiều người dùng hơn mà không cần phải phát triển lại từ đầu.

Unity cung cấp các công cụ và tùy chọn để tối ưu hóa hiệu suất và trải nghiệm người

dùng trên từng nền tảng, đảm bảo rằng trò chơi của bạn hoạt động mượt mà và ổn định.

4.1.2.6 Dịch vụ và Công cụ Hỗ trợ

Unity Editor tích hợp với nhiều dịch vụ và công cụ hỗ trợ khác nhau, bao gồm Unity Collaborate (hỗ trợ làm việc nhóm), Unity Analytics (theo dõi và phân tích dữ liệu người dùng), Unity Ads (quảng cáo trong game), Unity Multiplayer (phát triển game nhiều người chơi), và Unity Cloud Build (tự động xây dựng và phân phối bản build).

Các dịch vụ và công cụ này giúp bạn quản lý dự án một cách hiệu quả hơn, cung cấp các tính năng quan trọng để phát triển và phát hành trò chơi của mình.

4.1.3 Visual Studio

Visual Studio là một môi trường phát triển tích hợp (IDE) mạnh mẽ và phổ biến được phát triển bởi Microsoft. Được thiết kế để hỗ trợ các nhà phát triển trong việc xây dựng, gỡ lỗi và xuất bản các ứng dụng phần mềm trên nhiều nền tảng, Visual Studio cung cấp một bộ công cụ phong phú và toàn diện. Bài viết này sẽ trình bày chi tiết về Visual Studio, bao gồm các tính năng, công cụ hỗ trợ, và các phiên bản khác nhau của nó.

Một số tính năng nổi bật:

Hỗ trợ Đa Ngôn Ngữ: Visual Studio hỗ trợ nhiều ngôn ngữ lập trình khác nhau như C#, VB.NET, C++, JavaScript, TypeScript, Python, và nhiều ngôn ngữ khác thông qua các tiện ích mở rộng.

IntelliSense: Tính năng tự động hoàn thành mã giúp tăng tốc độ viết mã và giảm thiểu lỗi cú pháp. IntelliSense cung cấp gợi ý mã, thông tin về tham số hàm và tài liệu mã ngay trong quá trình viết.

Code Refactoring: Công cụ giúp cải thiện và tối ưu hóa mã nguồn bằng cách tái cấu trúc mã mà không làm thay đổi hành vi của ứng dụng. Bao gồm các thao tác như đổi tên biến, tách hàm, và hợp nhất các khối mã.

Debugging: Visual Studio cung cấp một bộ công cụ gỡ lỗi mạnh mẽ cho phép bạn đặt breakpoint, theo dõi biến, và kiểm tra ngăn xếp gọi hàm. Hỗ trợ gỡ lỗi trên cả ứng dụng cục bộ và từ xa.

Tích Hợp Git: Visual Studio tích hợp sẵn Git, cho phép bạn quản lý mã nguồn và cộng tác với các thành viên trong nhóm thông qua các nền tảng như GitHub, Azure Repos, và Bitbucket.

Source Control: Hỗ trợ các hệ thống kiểm soát phiên bản khác như Team Foundation Version Control (TFVC), giúp bạn theo dõi và quản lý các thay đổi mã nguồn một cách hiệu quả.

Unit Testing: Visual Studio cung cấp các công cụ để viết và chạy các bài kiểm tra đơn vị (unit tests) nhằm đảm bảo mã nguồn hoạt động chính xác.

4.2 Một số kết quả đạt được

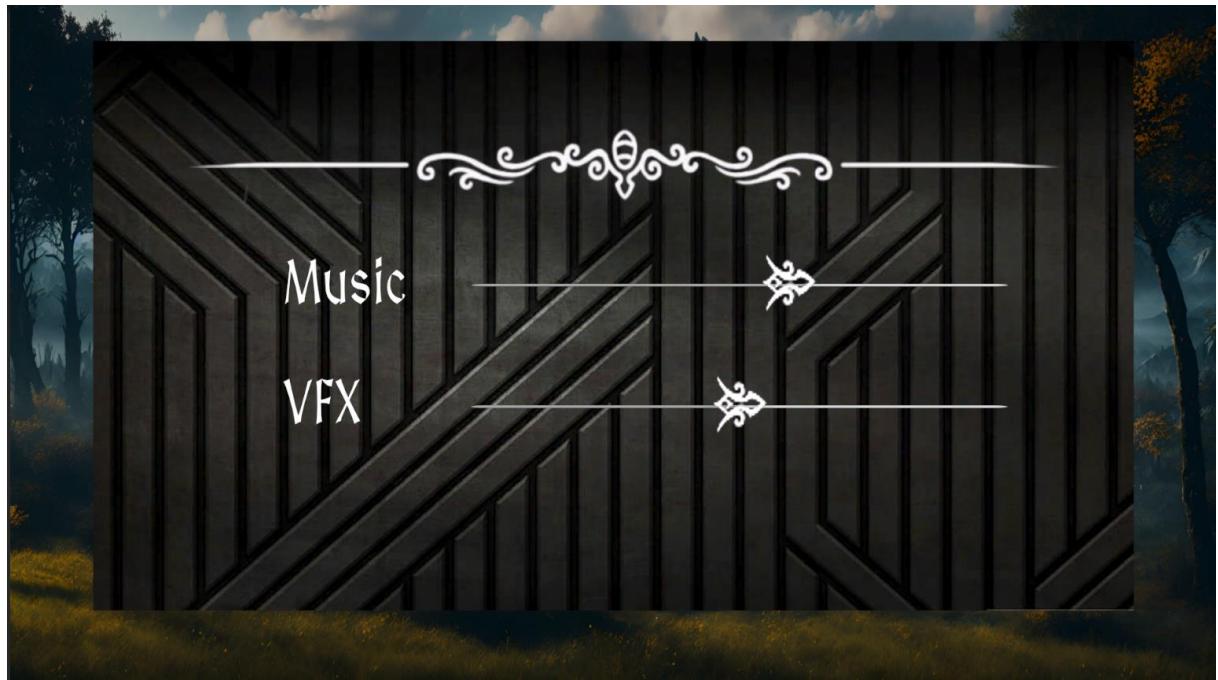
4.2.1 Giao diện tùy chọn của trò chơi

Khi người chơi khởi động trò chơi cửa sổ tùy chọn sẽ hiện lên. Người chơi có thể chọn “Start” để vào trò chơi



Hình 4.1 Menu chính

Nếu người chơi chọn “Setting” thì cửa sổ cài đặt âm thanh sẽ hiện lên cho phép người chơi tùy chỉnh âm thanh theo ý muốn.



Hình 4.2 Menu tùy chỉnh âm thanh

Sau khi chọn Start xong, cửa sổ tùy chỉnh chọn dữ liệu trò chơi theo ý muốn của người chơi. Nếu chọn “Continue” sẽ tiếp tục chơi dữ liệu của lần chơi trước. Nếu chọn “New Game” sẽ chơi lại từ đầu.



Hình 4.3 Menu chọn lưu trữ

Cửa sổ kết thúc trò chơi hiện ra sau khi người chơi hoàn thành trò chơi



Hình 4.4 Giao diện kết thúc trò chơi

4.2.1 Giao diện của trò chơi

Người chơi có thể di chuyển khám phá bản đồ.



Hình 4.5 Hình ảnh bản đồ

Người chơi có thể giao chiến với quái vật trên bản đồ.



Hình 4.6 Giao tranh với quái vật

Người chơi có thể tấn công vào một số bẫy nhất định để vượt qua địa hình.



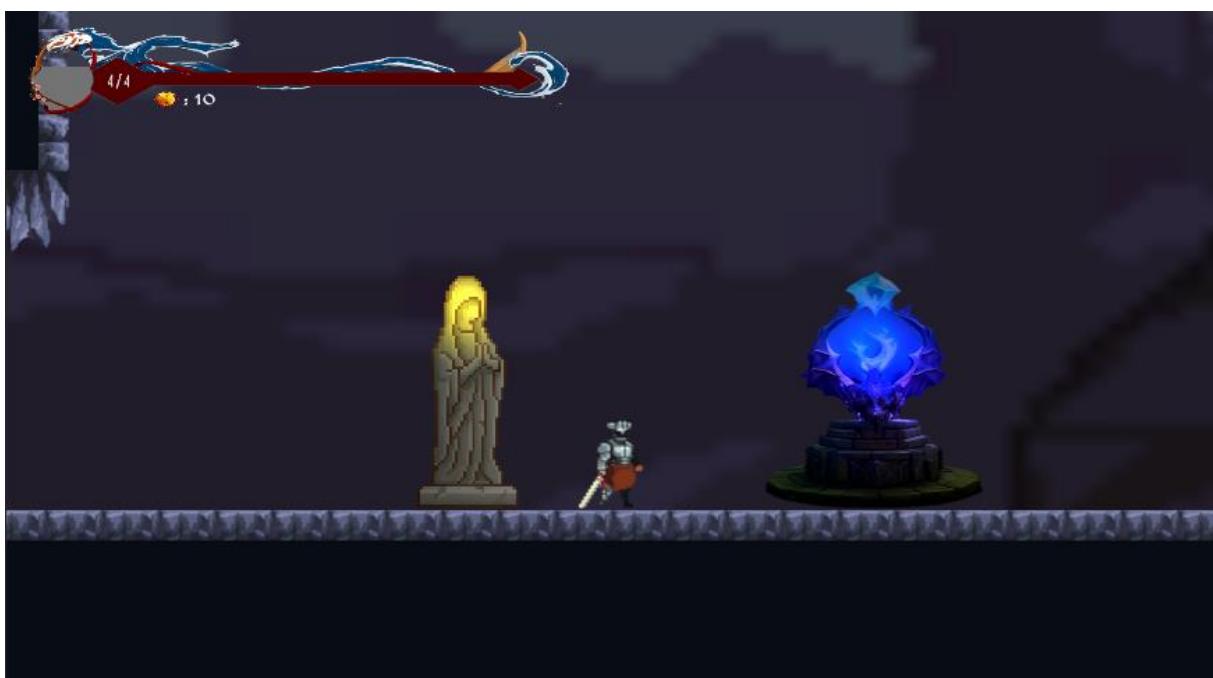
Hình 4.7 Vượt qua cạm bẫy

Người chơi có thể né tránh các đòn tấn công của quái vật.



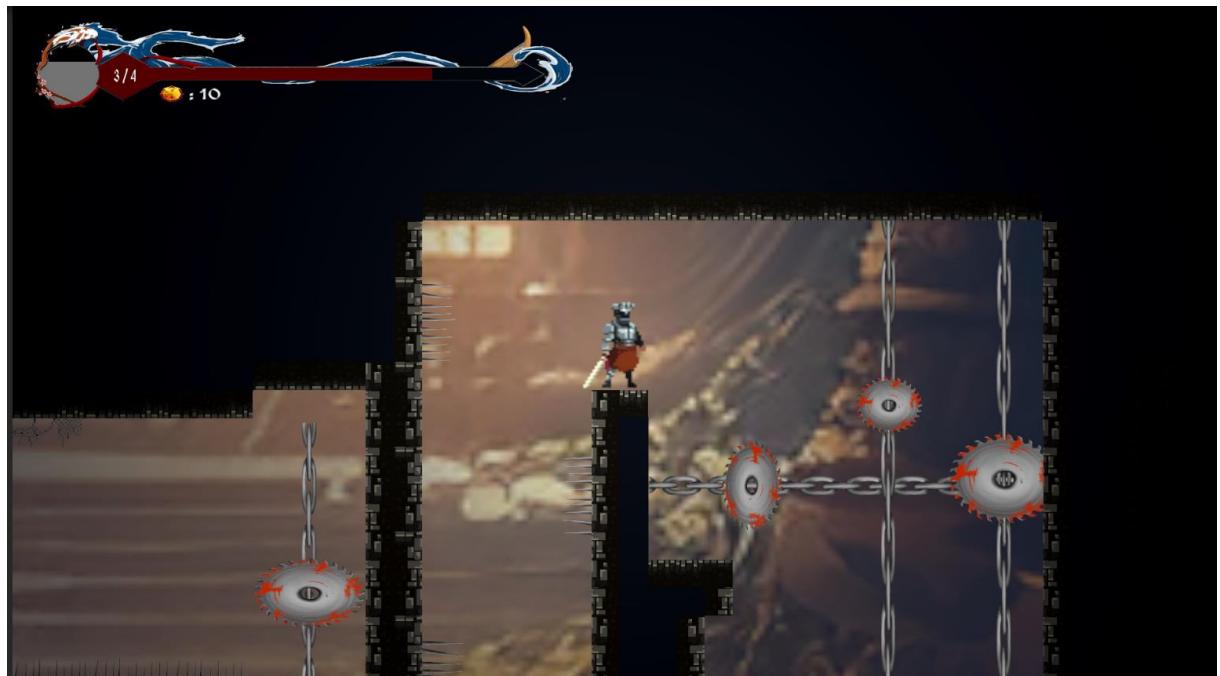
Hình 4.8 Né tránh đòn tấn công của quái vật

Người chơi đến điểm hồi sinh để hồi đầy máu và điểm kỹ năng để kiếm thêm kỹ năng mới.



Hình 4.9 CheckPoint và SkillPoint

Người chơi phải vượt qua các cạm bẫy nguy hiểm trong trò chơi.



Hình 4.10 Bẫy lưỡi cưa

Người chơi có thể ấn “ESC” trong lúc chơi để mở cửa sổ tùy chọn tạm dừng trò chơi.



Hình 4.10 Menu dừng trò chơi

4.3 Kết luận và hướng phát triển

4.3.1 Mục tiêu đạt được

Tựa game “The Knight Return” đã được xây dựng với đầy đủ các yếu tố:

Thiết kế giao diện game đơn giản dễ dàng sử dụng.

Cách chơi cơ bản dễ dàng tiếp cận.

Hệ thống điều khiển mượt mà, cấp độ khó tăng dần theo thời gian.

Game có cốt truyện, nội dung thú vị gây hứng thú với người chơi.

Tương thích trên nền tảng window đảm bảo hiệu suất tối ưu.

Game yêu cầu bảo mật dữ liệu tránh người dùng sửa đổi.

Hệ thống Save & Load lưu trữ tiến trình của người chơi.

4.3.2 Hạn chế

Mặc dù đã nỗ lực hết sức mình trong suốt quá trình làm đồ án, nhưng do giới hạn thời gian và lượng kinh nghiệm còn hạn chế nên cũng không thể tránh khỏi những thiếu sót như:

Lối chơi cơ bản chưa được đa dạng.

Nội dung của trò chơi còn sơ sài, chưa được phong phú.

Giao diện thiết kế chưa được đẹp mắt.

Chưa được kiểm thử nghiêm ngặt

4.4 Hướng phát triển

Mở rộng nội dung: Bổ sung thêm các khu vực mới để khám phá, các loại kẻ thù mới và những thử thách mới. Điều này giúp giữ cho người chơi luôn cảm thấy hứng thú và có động lực để tiếp tục hành trình.

Phát triển cốt truyện: Thêm các chương mới vào cốt truyện, mở rộng câu chuyện của

các nhân vật phụ và tạo ra các tuyến truyện phụ (side quests) để người chơi khám phá. Cải tiến đồ họa và âm thanh: Nâng cấp chất lượng đồ họa và âm thanh để tạo ra trải nghiệm hấp dẫn hơn. Điều này bao gồm việc cải thiện hiệu ứng hình ảnh, ánh sáng và âm thanh nền.

Cập nhật thường xuyên: Đưa ra các bản cập nhật thường xuyên để sửa lỗi, cải thiện trải nghiệm người chơi và bổ sung nội dung mới. Điều này giúp duy trì sự hứng thú của người chơi và đảm bảo trò chơi luôn được cải tiến.

TÀI LIỆU THAM KHẢO

- [1] Unity Documentation: <https://docs.unity3d.com/Manual/index.html>
- [2] Learn Unity: <https://learn.unity.com/>
- [3] Unity Tutorials: <https://unity3d.com/learn/tutorials>
- [4] C# Programming Guide: <https://learn.microsoft.com/en-us/dotnet/csharp/>
- [5] Unity Forums: <https://forum.unity.com/>
- [6] Unity Asset Store: <https://assetstore.unity.com/>