

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN

=====∞=====



ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

Đề tài:

Xây dựng Game chiến thuật phòng thủ 2D trên Unity Engine

GVHD : ThS. Trần Thanh Hùng

Sinh viên thực hiện : Nguyễn Văn Phương

Mã sinh viên : 2020603701

Lớp : CNTT03 – K15

Hà Nội - Năm 2024

LỜI CẢM ƠN

Trước tiên, em xin gửi lời cảm ơn đến ThS. Trần Thanh Hùng đã nhiệt tình hướng dẫn và giúp đỡ em trong thời gian làm đồ án để hoàn thành đề tài “Xây dựng Game chiến thuật phòng thủ 2D trên Unity Engine”. Sự hướng dẫn và những góp ý của thầy về kiến thức phân tích thiết kế hệ thống, cũng như những kiến thức thực tế và ứng dụng cao, đã giúp tôi phát triển không chỉ trong lĩnh vực này mà còn trong sự nghiệp chung.

Đặc biệt, trong quá trình thực hiện đề tài, thầy đã chỉ dẫn tôi cách phân tích và xây dựng cơ sở dữ liệu, hệ thống để đáp ứng nhu cầu của đề tài. Những buổi thảo luận sâu sắc đã giúp tôi hiểu sâu hơn về kiến thức và cách sửa các lỗi phát sinh trong quá trình thực hiện đề tài. Tôi đã có cơ hội nâng cao khả năng phân tích, thiết kế và tư duy để xây dựng hệ thống đạt chuẩn.

Cuốn đồ án này được hoàn thành theo đúng thời gian quy định của nhà trường cũng như của khoa không chỉ là sự nỗ lực của em mà còn sự giúp đỡ, chỉ bảo của thầy hướng dẫn.

Trong quá trình thực tập, cũng như là trong quá trình làm bài báo cáo đồ án, khó tránh khỏi sai sót, rất mong các thầy, cô bỏ qua. Đồng thời do trình độ lý luận cũng như kinh nghiệm thực tiễn còn hạn chế nên bài báo cáo không thể tránh khỏi những thiếu sót, em rất mong nhận được ý kiến đóng góp thầy, cô để em học thêm được nhiều kinh nghiệm và sẽ hoàn thành tốt nhất có thể!

Em xin chân thành cảm ơn!

MỤC LỤC

LỜI CẢM ƠN	2
DANH SÁCH CÁC TỪ VIẾT TẮT	8
DANH SÁCH CÁC BẢNG BIỂU	9
DANH SÁCH HÌNH VẼ.....	10
MỞ ĐẦU	12
CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI	13
1.1. Đặt vấn đề.....	13
1.2. Mục tiêu và phạm vi đề tài	13
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH HỆ THỐNG	15
2.1. Khảo sát hiện trạng.....	15
2.1.1. Các thể loại game phòng thủ phổ biến.....	15
2.1.2. Ưu điểm và nhược điểm.....	15
2.1.2.1. Tower Defense:	15
2.1.2.2. Hero Defense	16
2.1.2.3. Survival Defense	17
2.2. Phân tích hệ thống	19
2.2.1. Biểu đồ use case tổng quát.....	19
2.2.2. Biểu đồ phân rã use case chiến đấu	20
2.2.3. Biểu đồ phân rã use case vào kho chứa tháp	20
2.2.4. Biểu đồ phân rã use case vào cửa hàng.....	21
2.2.5. Đặc tả chức năng.....	21
2.2.5.1. Use case Chọn màn chơi.....	21
2.2.5.2. Use case Xây tháp.....	23

2.2.5.3. Use case Phá huỷ vật cản	23
2.2.5.4. Use case Đặt hỗ trợ	24
2.2.5.5. Use case Tăng sát thương	25
2.2.5.6. Use case Tăng phạm vi	26
2.2.5.7. Use case Tăng tốc độ bắn	26
2.2.5.8. Use case Vào cửa hàng	27
2.2.6. Đặc tả phi chức năng.....	28
2.2.6.1. Yêu cầu kỹ thuật	28
2.2.6.2. Yêu cầu lỗi chơi	28
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG	29
3.1. Ngôn ngữ lập trình C#.....	29
3.1.1. Giới thiệu.....	29
3.1.2. Điểm nổi bật.....	29
3.2. Mono Framework	30
3.3. Unity Engine.....	31
3.3.1. Khái niệm	31
3.3.2. Quá trình phát triển	31
3.3.3. Lý do chọn Unity Engine	31
3.4. Các thành phần quan trọng trong Unity	32
3.5. Công nghệ chính trong game.....	33
CHƯƠNG 4. KỊCH BẢN GAME.....	35
4.1. Bối cảnh trong game.....	35
4.2. Cốt truyện chính	35
4.3. Phong cảnh trong game	36

4.3.1.	Valley Vista (Thung Lũng)	36
4.3.2.	Desert Dunes (Sa Mạc)	37
4.3.3.	Urban Havoc (Đường Phố)	37
4.3.4.	Arctic Peaks (Bắc Cực).....	38
4.4.	Chi tiết các đối tượng trong game	38
4.4.1.	Tháp phòng thủ(quân ta).....	38
4.4.1.1.	Tháp Súng Máy(Machine Gun)	38
4.4.1.2.	Tháp Pháo(Cannon)	40
4.4.1.3.	Tháp Tầm xa(Snipper)	41
4.4.1.4.	Tháp Làm chậm(Time-Wrapper).....	42
4.4.1.5.	Tháp Laser(Laser).....	43
4.4.2.	Quân địch	44
4.4.3.	Chướng ngại vật trong game.....	45
4.4.4.	Hỗ trợ trong game	45
4.4.5.	Thiết kế level trong game.....	46
CHƯƠNG 5. THIẾT KẾ VÀ CÀI ĐẶT HỆ THỐNG		51
5.1.	Thiết kế kiến trúc.....	51
5.1.1.	Đối tượng (Object).....	51
5.1.2.	Lớp (Class).....	51
5.1.3.	Đặc tính cơ bản của OOP.....	51
5.2.	Thiết kế giao diện	52
5.2.1.	Thiết kế màn hình	52
5.2.2.	Màn hình Home	53
5.2.3.	Màn hình cài đặt.....	53

5.2.4.	Màn hình chọn Level	54
5.2.5.	Màn hình bắt đầu Level	54
5.2.6.	Màn hình Upgrade	55
5.2.7.	Màn hình trong game	55
5.2.8.	Màn hình tạm dừng game	56
5.2.9.	Màn hình chiến thắng.....	56
5.2.10.	Màn hình thua cuộc.....	57
5.3.	Thiết kế lớp.....	58
5.4.	Thiết kế cơ sở dữ liệu	60
5.5.	Công cụ sử dụng.....	64
5.6.	Kết quả đạt được.....	65
5.7.	Màn hình chức năng chính	65
5.7.1.	Màn hình Home	65
5.7.2.	Màn hình chọn màn chơi.....	66
5.7.3.	Màn hình cài đặt.....	66
5.7.4.	Màn hình thông tin game	67
5.7.5.	Màn hình nâng cấp tháp	67
5.7.6.	Màn hình map thung lũng	68
5.7.7.	Màn hình map sa mạc	68
5.7.8.	Màn hình map đường phố	69
5.7.9.	Màn hình map bắc cực	69
5.7.10.	Màn hình thua cuộc.....	70
5.7.11.	Màn hình chiến thắng.....	70
CHƯƠNG 6. KIỂM THỬ PHẦN MỀM.....		71

6.1.	Kế hoạch kiểm thử.....	71
6.2.	Kiểm thử chức năng	72
6.3.	Kiểm thử sức mạnh từng loại tháp trong game	73
CHƯƠNG 7. VẤN ĐỀ VÀ GIẢI PHÁP		74
7.1.	Vấn đề gặp phải	74
7.2.	Giải pháp	74
7.3.	Kết quả đạt được.....	74
CHƯƠNG 8. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN		75
8.1.	Kết luận	75
8.2.	Hướng phát triển.....	75
TÀI LIỆU THAM KHẢO		77

DANH SÁCH CÁC TỪ VIẾT TẮT

Từ viết tắt	Ý nghĩa
Enemy	Kẻ địch người sẽ tấn công căn cứ
GameObject	Thành phần đối tượng trong một cảnh của trò chơi.
GamePlay	GamePlay để chỉ những cách thức người chơi trải nghiệm game.
Map	Bản đồ
UI	Giao diện
Scene	Một cảnh trong game
Unity	Tên công cụ phát triển trò chơi

DANH SÁCH CÁC BẢNG BIỂU

Bảng 2.1. Biểu đồ use case chọn màn chơi.....	22
Bảng 2.2. Biểu đồ use case xây dựng tháp.....	23
Bảng 2.3. Biểu đồ use case phá hủy vật cản.	24
Bảng 2.4. Biểu đồ use case đặt hỗ trợ.....	25
Bảng 2.5. Biểu đồ use case tăng sát thương.....	25
Bảng 2.6. Biểu đồ use case tăng phạm vi.....	26
Bảng 2.7. Biểu đồ use case tăng tốc độ bắn.....	27
Bảng 2.8. Biểu đồ use case vào cửa hàng.....	28
Bảng 4.1. Thông tin cơ bản tháp súng máy	39
Bảng 4.2. Thông tin chi tiết tháp súng máy	39
Bảng 4.3. Thông tin cơ bản tháp pháo	40
Bảng 4.4. Thông tin chi tiết tháp pháo.....	40
Bảng 4.5. Thông tin cơ bản tháp súng ngắm	41
Bảng 4.6. Thông tin chi tiết tháp súng ngắm	41
Bảng 4.7. Thông tin cơ bản tháp làm chậm	42
Bảng 4.8. Thông tin chi tiết tháp làm chậm.....	42
Bảng 4.9. Thông tin cơ bản tháp laser	43
Bảng 4.10. Thông tin chi tiết tháp laser	43
Bảng 4.11. Thông tin chi tiết kẻ địch.....	44
Bảng 4.12. Thông tin chi tiết chướng ngại vật.....	45
Bảng 4.13. Thông tin chi tiết hỗ trợ.....	45
Bảng 4.14. Dữ liệu đường đi của từng level	50
Bảng 5.1. Thông tin màn hình.....	52
Bảng 5.2. Thông tin dung lượng dự án	65
Bảng 6.1. Thông tin máy tính kiểm thử	71
Bảng 6.2. Thông tin điện thoại kiểm thử	71
Bảng 6.3. Kiểm thử chức năng.....	73
Bảng 6.4. Kiểm thử sức mạnh tháp.....	73

DANH SÁCH HÌNH VẼ

Hình 2.1. Use case tổng quát.....	19
Hình 2.2. Use case phân rã chiến đấu	20
Hình 2.3. Use case vào kho chứa tháp	20
Hình 2.4. Use case vào cửa hàng	21
Hình 4.1. Map thung lũng	36
Hình 4.2. Map sa mạc	37
Hình 4.3. Map đường phố	37
Hình 4.4. Map bắc cực	38
Hình 5.1. Màn hình thiết kế home	53
Hình 5.2. Màn hình thiết kế cài đặt.....	53
Hình 5.3. Màn hình thiết kế chọn level.....	54
Hình 5.4. Màn hình thiết kế bắt đầu level.....	54
Hình 5.5. Màn hình thiết kế upgrade	55
Hình 5.6. Màn hình thiết kế trong game	55
Hình 5.7. Màn hình thiết kế tạm dừng game	56
Hình 5.8. Màn hình thiết kế chiến thắng.....	56
Hình 5.9. Màn hình thiết kế thua cuộc.....	57
Hình 5.10. Biểu đồ lớp.....	58
Hình 5.11. Biểu đồ trình tự nâng cấp tháp.....	59
Hình 5.12. Biểu đồ cơ sở dữ liệu	60
Hình 5.13. Biểu đồ lớp lưu trữ.....	61
Hình 5.14. Scriptable Object.....	62
Hình 5.15. Minh họa PlayerPrefs	63
Hình 5.16. Minh họa lưu dữ liệu trên scene.....	64
Hình 5.17. Màn hình home	65
Hình 5.18. Màn hình chọn màn chơi.....	66
Hình 5.19. Màn hình cài đặt.....	66
Hình 5.20. Màn hình thông tin game	67

Hình 5.21. Màn hình nâng cấp.....	67
Hình 5.22. Màn hình map thung lũng.....	68
Hình 5.23. Màn hình map sa mạc	68
Hình 5.24. Màn hình map đường phố.....	69
Hình 5.25. Màn hình map bắc cực.....	69
Hình 5.26. Màn hình thua cuộc.....	70
Hình 5.27. Màn hình chiến thắng.....	70

MỞ ĐẦU

Trong thời đại hiện nay, việc đáp ứng nhu cầu giải trí đã trở thành một việc tất yếu và quan trọng đối với mọi người. Tuy nhiên, trên thị trường game hiện nay vẫn chưa có nhiều tựa game có thể đáp ứng được nhu cầu của mọi người đặc biệt là những trò chơi có tính tập trung cao và yêu cầu độ chính xác và tính chiến thuật. Vì vậy, em quyết định phát triển một game 2D offline theo thể loại game chiến thuật phòng thủ để giải quyết vấn đề này. Game của em có cốt truyện hấp dẫn cùng với gameplay yêu cầu sự tập trung và chiến thuật cao. Người chơi sẽ xây dựng hệ thống phòng thủ để chống lại sự tấn công của kẻ thù. Nhiệm vụ đòi hỏi sự tập trung cao độ và tư duy chiến thuật của người chơi của người chơi khi phải đối mặt với những đợt tấn công dồn dập của kẻ địch trong nhiều trận chiến khốc liệt. Trong dự án này công cụ mà em lựa chọn là Unity Engine để xây dựng và phát triển game. Những tài nguyên trong trò chơi như hình ảnh, âm thanh, đồ họa sẽ được lấy từ hai nguồn là Unity Asset Store hoặc tự làm. Sau khi đã hoàn thành thì sản phẩm sẽ được triển khai trên nhiều nền tảng khác nhau như Desktop, mobile và website. Kết thúc đồ án sẽ là một sản phẩm nhằm đáp ứng được những nhu cầu giải trí của mọi người đặc biệt là những người đam mê những tựa game chiến thuật đòi hỏi sự tập trung cao độ. Tôi tin rằng dự án này sẽ đem lại nhiều trải nghiệm thú vị và ý nghĩa cho người chơi cũng như cộng đồng game thủ.

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1. Đặt vấn đề

Trong thời đại công nghệ thông tin ngày càng phát triển, game phòng thủ 2D đang trở thành một trong những thể loại được người chơi yêu thích và chú ý đặc biệt. Tuy nhiên, để tạo ra một trò chơi phòng thủ 2D mà không chỉ đạt được sự quan tâm của cộng đồng game thủ mà còn mang lại trải nghiệm tuyệt vời, đòi hỏi nhiều hơn chỉ là việc có kỹ năng lập trình và thiết kế đồ họa.

Vấn đề làm thế nào để phát triển một trò chơi phòng thủ 2D chất lượng, hấp dẫn và tương thích với nhiều thiết bị đang trở thành một thách thức lớn đối với các nhà phát triển. Để giải quyết được vấn đề này, đồ án tốt nghiệp này sẽ tập trung vào nghiên cứu sâu và phát triển một trò chơi phòng thủ 2D độc đáo và sáng tạo.

Mục tiêu của đồ án là không chỉ tạo ra một sản phẩm game chất lượng cao mà còn đáp ứng được mong muốn giải trí của người chơi. Việc tăng cường tính tương tác, động viên sự sáng tạo và tạo ra một thế giới game phong phú và hấp dẫn sẽ là trọng tâm của quá trình phát triển.

Qua việc áp dụng các phương pháp nghiên cứu và phân tích thị trường game hiện nay, cùng với sự sáng tạo và kỹ năng lập trình đỉnh cao, chúng tôi hy vọng sẽ đem lại một trải nghiệm game độc đáo, đầy lôi cuốn và thu hút sự quan tâm của đông đảo người chơi.

1.2. Mục tiêu và phạm vi đề tài

Tiếp theo em sẽ trình bày về mục tiêu và phạm vi đề tài của đồ án. Đầu tiên, em sẽ tổng quan lại các sản phẩm game phòng thủ 2D hiện tại và đánh giá các hạn chế của chúng. Hiện nay, trên thị trường đã có rất nhiều sản phẩm game phòng thủ 2D phong phú và đa dạng về chủ đề, đồ họa, âm thanh, tính năng, gameplay, v.v. Tuy nhiên, sau khi phân tích và đánh giá các sản phẩm này, chúng tôi nhận thấy vẫn tồn tại một số hạn chế nhất định.

Hạn chế về độ phong phú của nội dung: Game phòng thủ 2D thường có giới hạn về việc cung cấp nội dung mới và đa dạng. Mặc dù có thể thêm vào các cấp độ mới và vũ khí, nhưng sau một thời gian, người chơi có thể cảm thấy lặp lại.

Đồ họa hạn chế: Do hạn chế về không gian và cấu trúc, các game phòng thủ 2D thường có đồ họa đơn giản hơn so với game 3D. Điều này có thể làm giảm sự hấp dẫn và sự chân thực của trải nghiệm.

Thiếu tính tương tác: Mặc dù một số game phòng thủ 2D cung cấp sự tương tác với môi trường, nhưng thường thì nó không bằng với các game 3D. Sự hạn chế này có thể làm giảm sự tham gia của người chơi và cảm giác tham gia vào một thế giới sống động.

Khó khăn trong việc tạo ra trải nghiệm đa người chơi: Việc thiết kế các chế độ đa người chơi trong game phòng thủ 2D thường khó khăn hơn so với game 3D. Điều này có thể khiến cho trải nghiệm đa người chơi không được mạnh mẽ và hấp dẫn.

Giới hạn về khả năng kỹ thuật: Một số game phòng thủ 2D có thể gặp phải giới hạn về khả năng kỹ thuật của các thiết bị di động hoặc máy tính, điều này có thể ảnh hưởng đến hiệu suất và trải nghiệm chơi game.

Dựa trên các hạn chế trên, mục tiêu của đề tài là xây dựng một sản phẩm game phòng thủ 2D đáp ứng được mức độ tương tác và trải nghiệm người chơi cao, tính thách thức phù hợp và có thể trải nghiệm trên nhiều nền tảng khác nhau.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH HỆ THỐNG

Trong Chương 1, đã giới thiệu về mục tiêu, phạm vi và hướng đi của đề tài. Chương 2 sẽ có nhiệm vụ giới thiệu và so sánh các tựa game cùng thể loại đã có trên thị trường từ đó rút ra ưu và nhược điểm để phát triển, tiếp theo là những tính năng mà người chơi có thể thực hiện được trong game và phân tích chúng theo các biểu đồ của UML.

2.1. Khảo sát hiện trạng

2.1.1. Các thể loại game phòng thủ phổ biến

Game phòng thủ hiện nay được phân loại thành nhiều thể loại khác nhau tùy thuộc vào đồ họa, gameplay hình ảnh.

Tuy nhiên có 3 thể loại game phòng thủ phổ biến như:

1. Tower Defense (TD): Đây là thể loại game mà người chơi phải xây dựng các cấu trúc phòng thủ, thường là các tháp, để ngăn chặn việc quái vật hoặc kẻ thù tiến đến mục tiêu của họ. Ví dụ: Bloons Tower Defense 6, Kingdom rush, Toy defense.
2. Hero Defense: Trong các trò chơi này, người chơi điều khiển một nhóm các nhân vật có kỹ năng đặc biệt để đánh bại quái vật hoặc kẻ thù đang tấn công. Ví dụ: Plants vs. Zombies, Orcs Must Die! 2.
3. Survival Defense: Trò chơi này yêu cầu người chơi tồn tại qua các đợt tấn công liên tục từ quái vật hoặc mối đe dọa khác, thường bằng cách xây dựng và cải thiện căn cứ của họ. Ví dụ: 7 Days to Die, The Forest. [1]

2.1.2. Ưu điểm và nhược điểm

2.1.2.1. Tower Defense:

- **Ưu điểm:**

Dễ chơi, khó thống trị: Tower Defense thường có ngưỡng khó khăn tăng dần, tạo ra một trải nghiệm học và tiến bộ liên tục cho người chơi.

Chiến thuật đa dạng: Người chơi có thể thử nghiệm nhiều chiến lược và cấu hình tháp khác nhau để tìm ra phong cách chơi phù hợp nhất với họ.

Tiêu thụ thời gian hiệu quả: Các trận đấu thường ngắn gọn, phù hợp cho những người chơi có ít thời gian hoặc muốn trải nghiệm nhanh chóng.

- **Nhược điểm:**

Có thể trở nên lặp đi lặp lại: Một số trò chơi Tower Defense có ít sự đa dạng trong cơ chế và quái vật, làm cho trải nghiệm trở nên lặp đi lặp lại sau một thời gian.

Thiếu yếu tố hành động: So với Hero Defense hoặc Survival Defense, Tower Defense có ít yếu tố hành động trực tiếp, điều này có thể khiến một số người chơi cảm thấy thiếu kích thích.

2.1.2.2. *Hero Defense*

- **Ưu điểm:**

Sự đa dạng trong gameplay: Các nhân vật và kỹ năng đa dạng mang lại nhiều cách tiếp cận và chiến thuật cho người chơi.

Tính cá nhân hóa cao: Người chơi có thể lựa chọn nhân vật yêu thích và phát triển kỹ năng của họ theo hướng mong muốn.

Yếu tố hành động: Trong nhiều trò chơi Hero Defense, người chơi thường tham gia vào các trận đấu hành động trực tiếp, tạo ra trải nghiệm kịch tính và thú vị.

- **Nhược điểm:**

Phụ thuộc vào kỹ năng người chơi: Một số người chơi có thể gặp khó khăn trong việc điều khiển nhiều nhân vật hoặc quản lý các kỹ năng phức tạp.

Đòi hỏi kiên nhẫn và thời gian: Một số trận đấu có thể kéo dài và đòi hỏi người chơi phải đầu tư thời gian để hoàn thành.

2.1.2.3. Survival Defense

- **Ưu điểm:**

Tính tương tác với môi trường: Người chơi thường phải tìm kiếm tài nguyên, xây dựng căn cứ và tìm cách thích ứng với môi trường độc đáo của từng trò chơi.

Trải nghiệm thách thức: Việc phải đối mặt với các mối đe dọa nguy hiểm từ môi trường và quái vật tạo ra trải nghiệm thách thức và kịch tính.

Tính chiến lược: Người chơi phải lập kế hoạch và quản lý tài nguyên một cách cẩn thận để đảm bảo sự sống sót trong điều kiện khắc nghiệt.

- **Nhược điểm:**

Khả năng tái chơi giới hạn: Một số người chơi có thể cảm thấy lặp lại nếu không có đủ sự đa dạng trong các cơ sở hoặc môi trường.

Yêu cầu kiên nhẫn và thời gian: Xây dựng và duy trì một căn cứ mạnh mẽ có thể đòi hỏi thời gian và công sức đáng kể từ người chơi.

Dưới đây là một số lý do tại sao bạn có thể muốn chọn thể loại Tower Defense:

Trải nghiệm trò chơi đơn giản và dễ tiếp cận: Tower Defense thường có cơ chế trò chơi đơn giản, dễ hiểu và dễ chơi, phù hợp với cả người chơi mới bắt đầu và người chơi có kinh nghiệm.

Thời gian chơi linh hoạt: Các trận đấu Tower Defense thường ngắn gọn và có thể chơi trong thời gian ngắn, phù hợp với những người chơi có ít thời gian hoặc muốn trải nghiệm nhanh chóng.

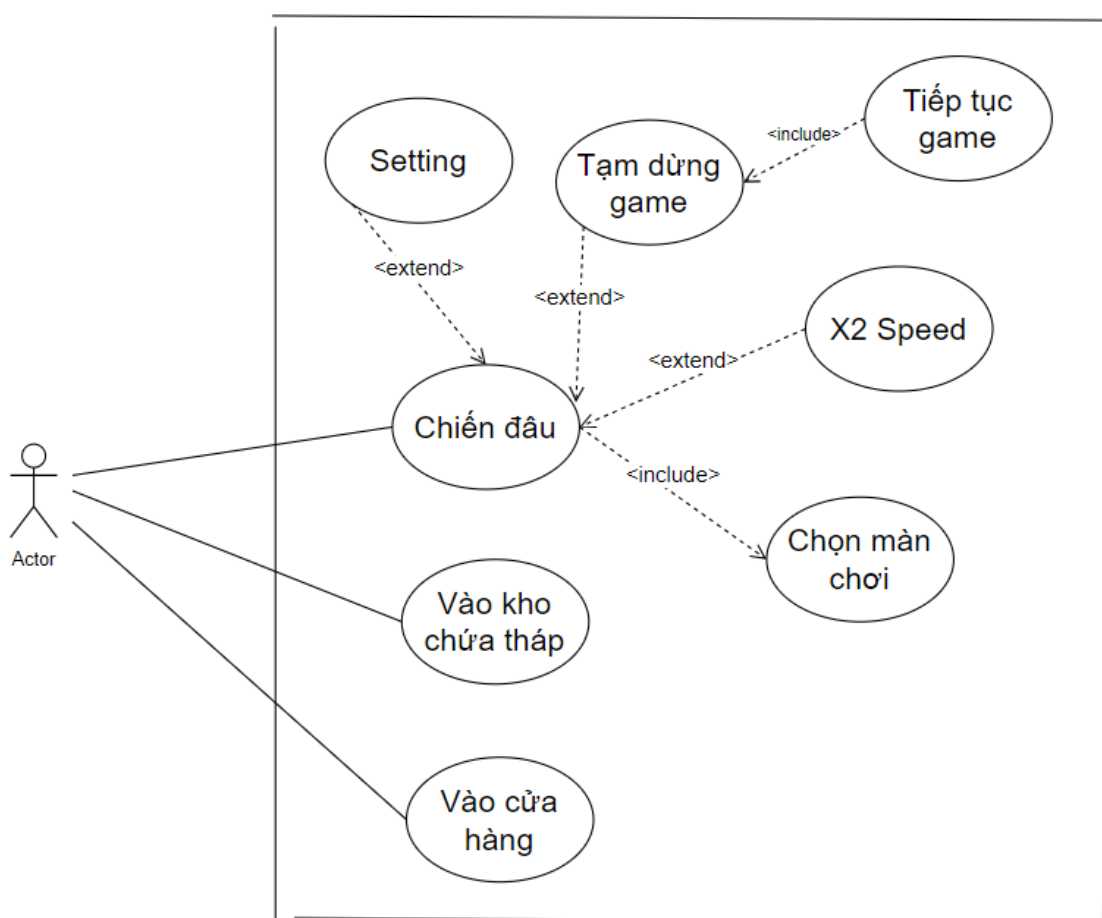
Tính chiến thuật và sáng tạo: Tower Defense yêu cầu người chơi phải lập kế hoạch và chiến thuật để đặt và nâng cấp các tháp phòng thủ một cách hiệu quả, tạo ra một môi trường thách thức và sáng tạo.

Thử nghiệm và cải thiện kỹ năng chiến lược: Việc thử nghiệm và điều chỉnh các chiến lược khác nhau trong Tower Defense giúp người chơi phát triển và cải thiện kỹ năng chiến thuật của mình.

Khả năng tương tác với cộng đồng: Các trò chơi Tower Defense thường có cộng đồng lớn, nơi bạn có thể chia sẻ kinh nghiệm, chiến lược và tham gia vào các thách đấu hoặc sự kiện cộng đồng.

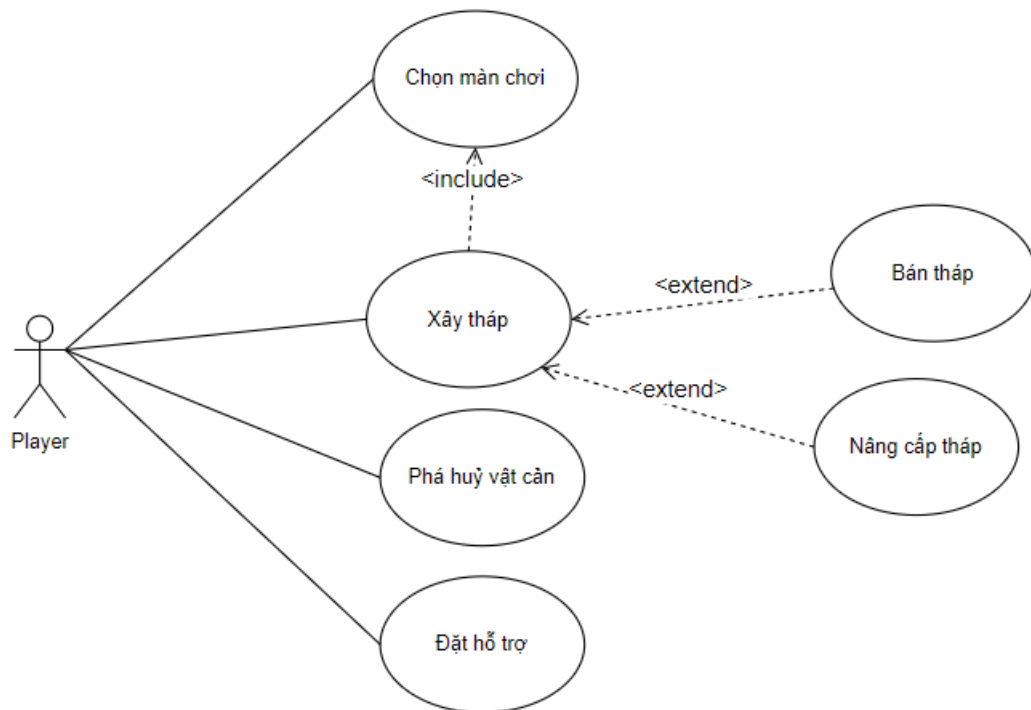
2.2. Phân tích hệ thống

2.2.1. Biểu đồ use case tổng quát



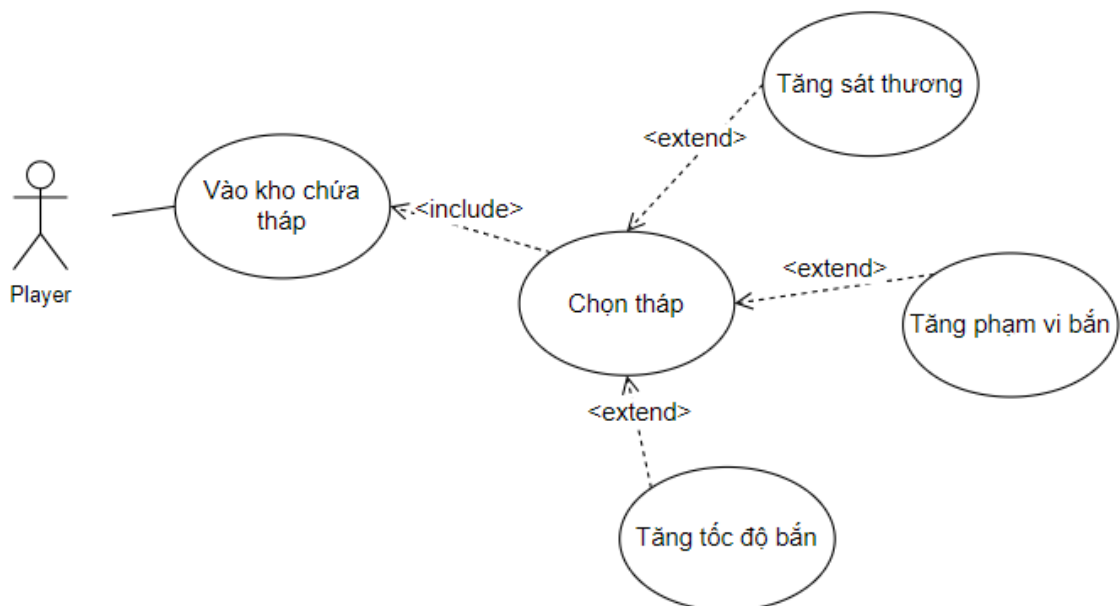
Hình 2.1. Use case tổng quát

2.2.2. Biểu đồ phân rã use case chiến đấu



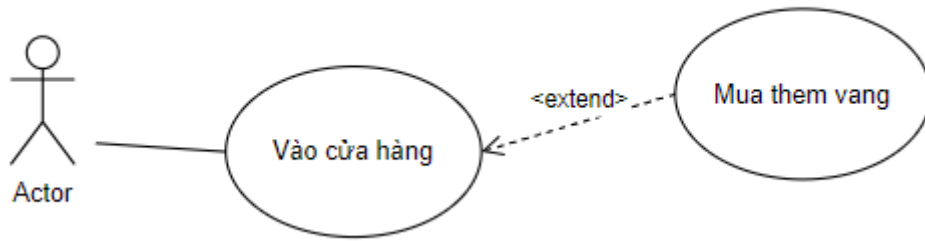
Hình 2.2. Use case phân rã chiến đấu

2.2.3. Biểu đồ phân rã use case vào kho chứa tháp



Hình 2.3. Use case vào kho chứa tháp

2.2.4. Biểu đồ phân rã use case vào cửa hàng



Hình 2.4. Use case vào cửa hàng

2.2.5. Đặc tả chức năng

2.2.5.1. Use case Chọn màn chơi

Use case “Chọn màn chơi”	
Tên usecase:	Chọn màn chơi
Mô tả:	Use case cho phép người chơi chọn một màn chơi để bắt đầu chiến đấu.
Luồng cơ bản:	<ol style="list-style-type: none"> 1. Người chơi: Người chơi kích vào nút xem danh sách màn chơi. 2. Hệ thống: Chuyển đến màn hình danh sách các màn chơi cho người chơi chọn. 3. Người chơi: Chọn một màn chơi. 4. Hệ thống: Hiện thị thông tin bao gồm: phần thưởng nhận được, số lượng tháp và loại tháp có thể đặt. 5. Người chơi: Ấn chọn vào màn chơi và chiến đấu. 6. Hệ thống: Chuyển sang scene chiến đấu.
Luồng rẽ nhánh:	Người chơi thoát khỏi màn hình chọn màn chơi.
Yêu cầu đặc biệt:	Không có.
Tiền điều kiện:	Người chơi phải qua màn trước đó để mở khoá màn tiếp theo.
Hậu điều kiện:	Người chơi hoàn thành màn chơi, nhận phần thưởng.

Điểm mở rộng:	Không có.
---------------	-----------

Bảng 2.1. Biểu đồ use case chọn màn chơi

2.2.5.2. Use case Xây tháp

Use case “Xây dựng tháp”	
Tên usecase:	Xây dựng tháp
Mô tả:	Use case cho phép người chơi xây dựng tháp để tấn công kẻ địch.
Luồng cơ bản:	<ol style="list-style-type: none"> 1. Người chơi: Người chơi kích vào một khoảng trống trên map. 2. Hệ thống: Hiện thị một PopUp gồm danh sách những tháp có thể xây dựng. 3. Người chơi: Chọn một tháp muốn xây dựng. 4. Hệ thống: Xây dựng tháp và trừ tiền.
Luồng rẽ nhánh:	<ol style="list-style-type: none"> 1. Người chơi thoát Popup xây dựng tháp 2. Người chơi nâng cấp tháp 3. Người chơi bán tháp đã có
Yêu cầu đặc biệt:	Không có.
Tiền điều kiện:	Người chơi phải đủ tiền để mua tháp. Người chơi chọn đúng vị trí không có vật cản trên map.
Hậu điều kiện:	Người chơi xây dựng được tháp.
Điểm mở rộng:	Không có.

Bảng 2.2. Biểu đồ use case xây dựng tháp

2.2.5.3. Use case Phá huỷ vật cản

Use case “Phá huỷ chướng ngại vật”	
Tên usecase:	Phá huỷ vật cản
Mô tả:	Use case cho phép người chơi phá huỷ vật cản để xây dựng tháp.
Luồng cơ bản:	<ol style="list-style-type: none"> 5. Người chơi: Người chơi kích vào một chướng ngại vật muốn phá huỷ.

	6. Hệ thống: Điều khiển những tháp có thể tấn công trướng ngại vật để phá hủy chướng ngại vật.
Luồng rẽ nhánh:	Trận đấu kết thúc trong khi đang phá hủy chướng ngại vật.
Yêu cầu đặc biệt:	Không có.
Tiền điều kiện:	Người chơi phải xây dựng tháp trước đó.
Hậu điều kiện:	Người chơi phá hủy được chướng ngại vật.
Điểm mở rộng:	Không có.

Bảng 2.3. Biểu đồ use case phá hủy vật cản.

2.2.5.4. Use case Đặt hỗ trợ

Use case “Đặt hỗ trợ”	
Tên usecase:	Đặt hỗ trợ
Mô tả:	Use case cho phép người chơi đặt một hỗ trợ vào đường đi của kẻ địch.
Luồng cơ bản:	<ol style="list-style-type: none"> 1. Người chơi: Người chơi kích vào đường đi của địch. 2. Hệ thống: Hiện thị Popup gồm danh sách những ability 3. Người chơi: Chọn một ability để sử dụng.
Luồng rẽ nhánh:	Trận đấu kết thúc trong khi đang phá hủy chướng ngại vật.
Yêu cầu đặc biệt:	Không có.
Tiền điều kiện:	Ability phải đủ không trong thời gian hồi chiêu

Hậu điều kiện:	Người chơi sử dụng được ability.
Điểm mở rộng:	Không có.

Bảng 2.4. Biểu đồ use case đặt hỗ trợ

2.2.5.5. Use case Tăng sát thương

Use case “Tăng sát thương”	
Tên usecase:	Tăng sát thương
Mô tả:	Use case cho phép người chơi gia tăng sát thương của tháp trong kho chứa.
Luồng cơ bản:	<ol style="list-style-type: none"> 1. Người chơi: Người chơi kích vào một loại tháp muốn gia tăng sát thương. 2. Hệ thống: Hiện thị Popup gồm thông tin nâng cấp. 3. Người chơi: Người chơi nhấn đồng ý nâng cấp hoặc không. 4. Hệ thống: Nâng cấp tháp theo lựa chọn của người chơi.
Luồng rẽ nhánh:	Tháp được chọn đã đạt sát thương tối đa có thể nâng cấp.
Yêu cầu đặc biệt:	Không có.
Tiền điều kiện:	Người chơi phải mở khoá tháp muốn nâng cấp trước để có thể nâng cấp.
Hậu điều kiện:	Sát thương của tháp được cập nhật.
Điểm mở rộng:	Không có.

Bảng 2.5. Biểu đồ use case tăng sát thương

2.2.5.6. Use case Tăng phạm vi

Use case “Tăng phạm vi”	
Tên usecase:	Tăng phạm vi
Mô tả:	Use case cho phép người chơi gia tăng phạm vi tấn công của tháp trong kho chứa
Luồng cơ bản:	<ol style="list-style-type: none"> 1. Người chơi: Người chơi kích vào một loại tháp muốn gia tăng phạm vi tấn công. 2. Hệ thống: Hiện thị Popup gồm thông tin nâng cấp. 3. Người chơi: Người chơi nhấn đồng ý nâng cấp hoặc không. <p>Hệ thống: Nâng cấp tháp theo lựa chọn của người chơi.</p>
Luồng rẽ nhánh:	Tháp được chọn đã đạt phạm vi tối đa có thể nâng cấp.
Yêu cầu đặc biệt:	Không có.
Tiền điều kiện:	Người chơi phải mở khoá tháp muốn nâng cấp trước để có thể nâng cấp.
Hậu điều kiện:	Phạm vi của tháp được cập nhật.
Điểm mở rộng:	Không có.

Bảng 2.6. Biểu đồ use case tăng phạm vi

2.2.5.7. Use case Tăng tốc độ bắn

Use case “Tăng tốc độ bắn”	
Tên usecase:	Tăng tốc độ bắn
Mô tả:	Use case cho phép người gia tăng tốc độ bắn của tháp trong kho chứa.

Luồng cơ bản:	<ol style="list-style-type: none"> 1. Người chơi: Người chơi kích vào đường đi của địch. 2. Hệ thống: Hiện thị Popup gồm danh sách những ability 3. Người chơi: Chọn một ability để sử dụng.
Luồng rẽ nhánh:	Tháp được chọn đã đạt phạm vi tối đa có thể nâng cấp.
Yêu cầu đặc biệt:	Không có.
Tiền điều kiện:	Tốc độ bắn của tháp được cập nhật
Hậu điều kiện:	Người chơi sử dụng được ability.
Điểm mở rộng:	Không có.

Bảng 2.7. Biểu đồ use case tăng tốc độ bắn

2.2.5.8. Use case Vào cửa hàng

Use case “Vào cửa hàng”	
Tên usecase:	Vào cửa hàng
Mô tả:	Use case cho phép người chơi mua thêm vàng.
Luồng cơ bản:	<ol style="list-style-type: none"> 1. Người chơi: Người chơi kích vào nút cửa hàng. 2. Hệ thống: Hiện thị Popup gồm danh sách những tùy chọn để người chơi mua. 3. Người chơi: Chọn một tùy chọn để mua.
Luồng rẽ nhánh:	Người chơi rời khỏi shop hoặc không đủ tiền.
Yêu cầu đặc biệt:	Không có.
Tiền điều kiện:	Người chơi phải có đủ tiền để mua

Hậu điều kiện:	Người chơi mua thêm được vàng.
Điểm mở rộng:	Không có.

Bảng 2.8. Biểu Biểu đồ use case vào cửa hàng

2.2.6. Đặc tả phi chức năng

2.2.6.1. Yêu cầu kỹ thuật

Hệ thống điều khiển trong game phải nhanh mượt mà tốc độ di chuyển, thay đổi view camera, nút bấm, nút reset...

- **Đồ họa:**

Sử dụng đồ họa hợp lý là vô cùng quan trọng trong game phải đảm bảo được những yếu tố như bắt mắt thu hút người chơi nhưng vẫn ổn định về mặt hiệu suất của game.

- **Âm thanh:**

Âm thanh trong game là vô cùng quan trọng làm tăng tính chân thật và sống động hơn trong game, phải chọn âm thanh phù hợp với những sự kiện và tình huống cụ thể trong game.

- **Bảo mật:**

Tính bảo mật là yếu tố mà game nào cũng cần được chú trọng, để tránh việc người chơi có thể hack dữ liệu dẫn đến việc mất cân bằng game

2.2.6.2. Yêu cầu lối chơi

Lối chơi trong game phải đa dạng và phong phú, độ khó nên tăng dần theo từng cấp độ để tránh việc người chơi cảm thấy sự lặp lại nhàm chán.

CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

Trong chương trước đã phân tích chi tiết hệ thống bao gồm các use case chính trong game. Chương này sẽ giới thiệu về công nghệ mà game sẽ sử dụng để phát triển.

3.1. Ngôn ngữ lập trình C#

3.1.1. Giới thiệu

Ngôn ngữ lập trình C# là một ngôn ngữ lập trình đa mục đích được Microsoft phát triển từ năm 2000. Dựa trên các tiêu chuẩn và cú pháp của ngôn ngữ C, C# được thiết kế để cung cấp một môi trường lập trình an toàn, hiệu quả và linh hoạt cho việc phát triển ứng dụng trên nền tảng Microsoft. Dưới đây là một số điểm nổi bật về C#. [2]

3.1.2. Điểm nổi bật

- **Dễ học và sử dụng:** C# được thiết kế với cú pháp gần gũi với ngôn ngữ tự nhiên, làm cho việc học và sử dụng nó dễ dàng hơn đối với người mới bắt đầu và các nhà phát triển có kinh nghiệm.
- **Tính đa năng:** C# hỗ trợ nhiều loại ứng dụng khác nhau, từ ứng dụng desktop đến ứng dụng web và di động. Bạn có thể sử dụng C# để phát triển ứng dụng trên nền tảng Windows, macOS và Linux.
- **Tính đồng nhất:** C# được tích hợp chặt chẽ với các công cụ và thư viện của Microsoft, giúp việc phát triển ứng dụng trở nên mạnh mẽ và dễ dàng hơn.
- **An toàn:** C# có các tính năng như quản lý bộ nhớ tự động và kiểu dữ liệu an toàn, giúp tránh được các lỗi như tràn bộ nhớ và lỗi trỏ.
- **Hỗ trợ mạnh mẽ:** C# có một cộng đồng lớn và sôi động, cùng với tài liệu phong phú và các công cụ hỗ trợ, giúp việc phát triển ứng dụng trở nên dễ dàng hơn.

- **Kết hợp tốt:** C# kết hợp tốt với nền tảng .NET, cho phép bạn sử dụng một loạt các thư viện và khung làm việc để phát triển các ứng dụng phức tạp.

3.2. Mono Framework

Mono là một framework mã nguồn mở được sử dụng trong Unity để hỗ trợ việc phát triển ứng dụng và trò chơi bằng ngôn ngữ lập trình C#. Đây là một phần quan trọng trong cơ sở hạ tầng của Unity và mang lại nhiều lợi ích cho nhà phát triển. Dưới đây là một số điểm quan trọng về Mono framework trong Unity:

- **Ngôn ngữ lập trình C#:** Mono framework hỗ trợ việc sử dụng ngôn ngữ lập trình C# trong Unity. C# là một ngôn ngữ phổ biến và mạnh mẽ, cung cấp cú pháp dễ đọc và hiểu, cũng như các tính năng mạnh mẽ như quản lý bộ nhớ tự động và kiểu dữ liệu an toàn.
- **Cross-platform:** Mono framework cho phép bạn viết mã một lần và chạy trên nhiều nền tảng khác nhau mà không cần phải viết lại mã nguồn. Điều này giúp giảm thời gian và công sức cần thiết để phát triển và triển khai ứng dụng trên nhiều thiết bị và hệ điều hành.
- **Performance:** Mono được tối ưu hóa để cung cấp hiệu suất tốt trong việc thực thi mã nguồn C#. Các cải tiến liên tục được thực hiện để tăng cường hiệu suất và tối ưu hóa tiêu thụ tài nguyên của ứng dụng, giúp trò chơi chạy mượt mà và không bị giật lag.
- **Tiện ích tích hợp:** Mono framework được tích hợp chặt chẽ với Unity Editor, giúp bạn dễ dàng tạo, chỉnh sửa và quản lý mã nguồn C# của mình trong quá trình phát triển trò chơi và ứng dụng. [3]

3.3. Unity Engine

3.3.1. Khái niệm

Unity là một “cross- platform game engine” tạm hiểu là công cụ phát triển game đa nền tảng được phát triển bởi Unity Technologies. Game engine này được sử dụng để phát triển game trên PC, consoles, thiết bị di động và trên websites.

3.3.2. Quá trình phát triển

Ra mắt đầu tiên vào năm 2005 tại sự kiện Apple’s Worldwide Developer Conference bởi nhà sáng lập David Helgason, trải qua hơn 12 năm phát triển, nay Unity đã có version 5.5 hoàn thiện hơn về rất nhiều mặt. Tháng 5-2012 theo cuộc khảo sát Game Developer Megazine được công nhận là Game engine tốt nhất cho mobile. Năm 2014 Unity thắng giải “Best Engine” tại giải UK’s annual Develop Industry Excellence.

3.3.3. Lý do chọn Unity Engine

1. **Đa nền tảng:** Unity hỗ trợ phát triển ứng dụng trên nhiều nền tảng khác nhau như PC, điện thoại di động (iOS, Android), máy tính bảng, console game, VR và AR.
2. **Dễ sử dụng:** Unity cung cấp một giao diện đồ họa thân thiện và dễ sử dụng, giúp người dùng nhanh chóng làm quen và bắt đầu phát triển trò chơi và ứng dụng của riêng họ.
3. **Khả năng tùy biến cao:** Unity cho phép người dùng tùy chỉnh và mở rộng chức năng của nó thông qua các plugin và scripting. Điều này giúp các nhà phát triển tạo ra các trải nghiệm duyệt web, đồ họa chất lượng cao, và hệ thống game phức tạp.
4. **Hỗ trợ đa dạng:** Unity cung cấp nhiều tài liệu học tập, video hướng dẫn, và diễn đàn trực tuyến để hỗ trợ người dùng trong quá trình phát triển ứng dụng của họ.
5. **Tiện ích kỹ thuật:** Unity cung cấp các công cụ kỹ thuật như physics engine, animation system, và audio engine, giúp người dùng dễ dàng tạo ra các trò chơi và ứng dụng đẹp mắt và chất lượng cao.

3.4. Các thành phần quan trọng trong Unity

1. **Scene (Cảnh):** Một Scene là một không gian 3D hoặc 2D chứa tất cả các đối tượng, hiệu ứng, ánh sáng và các phần tử khác của trò chơi hoặc ứng dụng của bạn. Trong một project, bạn có thể có nhiều scene khác nhau.
2. **Game Object (Đối tượng trò chơi):** Game Object là thành phần cơ bản nhất trong Unity, đại diện cho mọi thứ trong trò chơi hoặc ứng dụng của bạn, từ nhân vật, vật phẩm, đèn ánh sáng và camera. Mỗi Game Object có các thuộc tính như vị trí, quay, tỉ lệ, và các thành phần liên quan khác.
3. **Component (Thành phần):** Mỗi Game Object trong Unity được xây dựng từ các thành phần (Component). Các thành phần này định nghĩa hành vi và tính năng của Game Object. Ví dụ, một Game Object có thể có thành phần Collider để xác định vùng va chạm, hoặc có thành phần Renderer để hiển thị hình ảnh.
4. **Script (Kịch bản):** Script là các đoạn mã được viết bằng ngôn ngữ lập trình như C# hoặc JavaScript và gắn với Game Object để điều khiển hành vi của chúng. Script giúp bạn tạo ra các chức năng tùy chỉnh và xử lý logic trong trò chơi hoặc ứng dụng của mình.
5. **Prefab:** Prefab là một mẫu Game Object hoặc một nhóm Game Object có thể được tái sử dụng trong các scene khác nhau. Prefab giúp bạn tạo ra và quản lý các đối tượng có cấu trúc phức tạp một cách dễ dàng.
6. **Asset:** Asset là mọi thứ mà bạn thêm vào dự án của mình như hình ảnh, âm thanh, video, các tài nguyên 3D, v.v. Asset giúp bạn tạo ra các nội dung đa dạng và phong phú cho trò chơi hoặc ứng dụng của mình.
7. **Camera (Máy ảnh):** Camera đại diện cho góc nhìn của người chơi trong trò chơi. Nó quyết định cái gì được hiển thị trên màn hình và cách mà các đối tượng di chuyển trong không gian.
8. **Lighting (Ánh sáng):** Lighting quản lý cách ánh sáng được phản chiếu và chiếu sáng trong trò chơi hoặc ứng dụng của bạn. Unity hỗ trợ nhiều

loại ánh sáng và hiệu ứng ánh sáng khác nhau để tạo ra môi trường sống động và hấp dẫn.

3.5. Công nghệ chính trong game

Physics Engine trong Unity là một phần quan trọng của việc tạo ra các trò chơi và ứng dụng có tính tương tác vật lý. Đây là một công nghệ mạnh mẽ giúp mô phỏng các hiện tượng vật lý như đối lực, va chạm, đàn hồi, ma sát và nhiều hơn nữa. Dưới đây là một số khía cạnh chi tiết về bộ vật lý trong Unity:

1. **Rigidbody:** Rigidbody là một thành phần quan trọng trong Unity được sử dụng để áp dụng các luật vật lý lên các đối tượng. Khi một Game Object có Rigidbody, nó sẽ được ảnh hưởng bởi các lực như lực hấp dẫn, lực va chạm và lực đàn hồi. Rigidbody cho phép các đối tượng di chuyển, quay, và phản ứng với các lực mô phỏng.
2. **Collider:** Collider là một thành phần khác được sử dụng để xác định khu vực va chạm của một đối tượng. Khi hai Collider va chạm vào nhau, bộ vật lý trong Unity sẽ tính toán các phản ứng va chạm như va đập và lực đẩy. Collider có thể là các hình dạng đa dạng như hình cầu, hình hộp, hay mesh collider.
3. **Áp dụng lực và tác động:** Unity cung cấp các phương thức để áp dụng lực và tác động lên các đối tượng trong trò chơi. Bằng cách sử dụng các hàm như `AddForce()` và `AddTorque()`, bạn có thể điều khiển việc di chuyển và quay của các đối tượng dựa trên các tác động vật lý.
4. **Materials:** Trong Unity, bạn có thể xác định các vật liệu (Materials) cho các Collider để xác định các thuộc tính vật lý như độ đàn hồi, ma sát và cản trở. Điều này cho phép bạn tùy chỉnh cách mà các đối tượng phản ứng với nhau trong môi trường vật lý.
5. **TileMap:** Đây là công cụ giúp người chơi vẽ ra những hình ảnh, đối tượng sử dụng bút để chấm trên một lưới. [4]

Collision Detection (Phát hiện va chạm): Unity cung cấp các phương pháp để phát hiện và xử lý các va chạm giữa các Collider. Bằng cách sử dụng các hàm như `OnCollisionEnter()`, `OnCollisionStay()`, và `OnCollisionExit()`, bạn có thể thực hiện các hành động tùy thuộc vào việc các đối tượng chạm vào nhau.

CHƯƠNG 4. KỊCH BẢN GAME

Chương trước đã giới thiệu về công nghệ sử dụng, ở chương này sẽ phân tích về bối cảnh, cốt chuyện kịch bản và thông số chi tiết trong game.

4.1. Bối cảnh trong game

Trên một vùng đất bí ẩn, nơi mà thiên nhiên còn nguyên vẹn và tài nguyên phong phú, một cuộc đối đầu lớn giữa hai phe đã nổ ra. Tại đây, một lực lượng đại diện cho sự bảo vệ và bền vững, gọi là "Alliance of Nations" (Liên minh các quốc gia), đối đầu với một tập đoàn lợi ích cá nhân được biết đến với cái tên "Greed Corporation" (Tập đoàn Tham lam). Cuộc chiến này không chỉ đặt ra câu hỏi về quyền lợi và quyền tự do, mà còn đề cập đến vấn đề của sự sống và sự tồn vong của môi trường tự nhiên.

4.2. Cốt truyện chính

Vùng đất phức tạp và đa dạng này đã trở thành tâm điểm của cuộc chiến tranh đẫm máu giữa Alliance of Nations và Greed Corporation. Tại đây, mỗi bên đã triệu tập những chiến binh và lính của mình, sẵn sàng chiến đấu để bảo vệ hoặc chiếm đóng vùng đất quan trọng này. Tuy nhiên, mỗi bên đều có một mục tiêu khác nhau: Alliance of Nations mong muốn bảo vệ và sử dụng tài nguyên một cách bền vững, trong khi Greed Corporation chỉ quan tâm đến việc khai thác tài nguyên để lợi ích cá nhân mà không quan tâm đến hậu quả môi trường. [5]

Người chơi sẽ vào vai một chỉ huy tài ba, phải lãnh đạo lực lượng của mình thông qua những trận chiến khốc liệt để đối phó với các cuộc tấn công của Greed Corporation và bảo vệ vùng đất quý giá này. Họ sẽ phải sử dụng tất cả các kỹ năng của mình để chiến đấu thông minh và hiệu quả trên các địa hình đa dạng, từ rừng rậm sâu thẳm đến núi non cao vút và sa mạc cằn cỗi.

Mỗi trận đấu sẽ đòi hỏi người chơi phải đưa ra những quyết định chiến lược thông minh, sử dụng tài nguyên một cách hiệu quả và xây dựng hệ thống phòng thủ vững chắc. Họ cũng sẽ phải đối mặt với những quyết định đạo đức

khó khăn, khi phải cân nhắc giữa việc bảo vệ môi trường và chiến đấu cho lợi ích của họ. Đồng thời, họ cũng sẽ phải khám phá những bí ẩn về vùng đất này, nhằm đảm bảo rằng sự cân bằng tự nhiên sẽ được duy trì và môi trường sẽ được bảo tồn cho thế hệ tương lai.

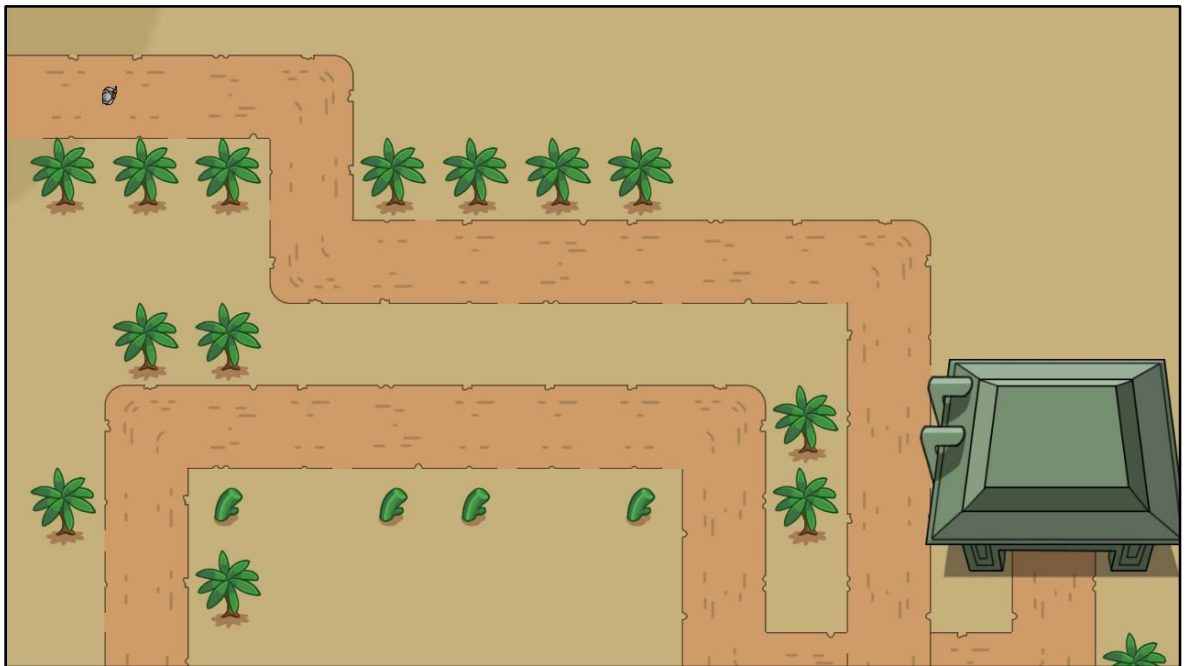
4.3. Phong cảnh trong game

4.3.1. Valley Vista (Thung Lũng)



Hình 4.1. Map thung lũng

4.3.2. Desert Dunes (Sa Mạc)



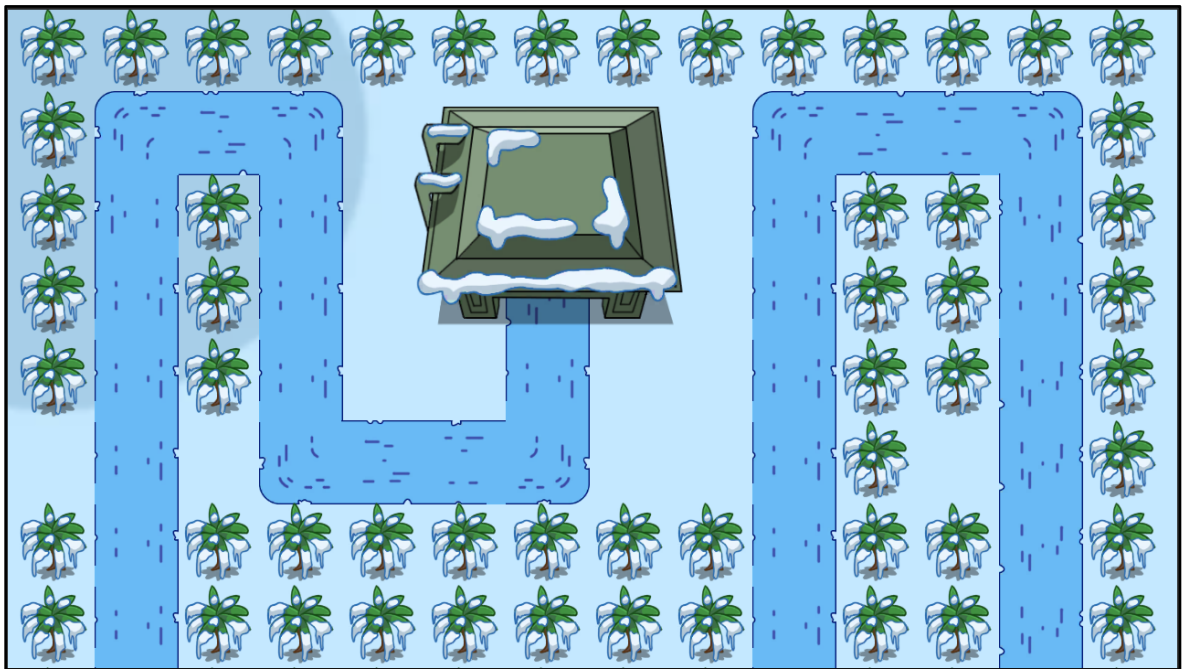
Hình 4.2. Map sa mạc

4.3.3. Urban Havoc (Đường Phố)



Hình 4.3. Map đường phố

4.3.4. Arctic Peaks (Bắc Cực)



Hình 4.4. Map bắc cực

4.4. Cách chơi


Người chơi có thể chọn những vùng đất có khoảng trống trên bản đồ bằng cách click chuột trái (Trên Desktop) hoặc chạm (Trên điện thoại) để mở giao diện tùy chọn tháp muốn đặt. Sau đó người chơi chọn tháp muốn đặt để xây dựng.



4.5. Chi tiết các đối tượng trong game

4.5.1. Tháp phòng thủ (quân ta)

4.5.1.1. Tháp Súng Máy (Machine Gun)

1) Thông tin cơ bản

Cấp độ	Tên	Mô Tả	Hình ảnh
1	MG-1	Phạm vi: Súng máy bắn đơn mục tiêu. Số nòng: 1	

2	MG-2	Phạm vi:Súng máy bắn đơn mục tiêu. Số nòng: 1	
3	MG-3	Phạm vi:Súng máy bắn đơn mục tiêu. Số nòng: 2	

Bảng 4.1. Thông tin cơ bản tháp súng máy




2) Thông tin chi tiết

Cấp độ	Sát thương mỗi lần bắn	Tỉ lệ bắn	Phạm vi	Giá xây	Giá bán
1	10	0.25 giây	1	100 \$	75 \$
2	20	0.25 giây	1.5	200 \$	150 \$
3	40	0.125 giây	2.25	300 \$	225 \$

Bảng 4.2. Thông tin chi tiết tháp súng máy

4.5.1.2. Tháp Pháo(Cannon)

1) Thông tin cơ bản

Cấp độ	Tên	Mô Tả	Hình ảnh
1	Single Cannon	Phạm vi:Súng máy bắn đơn mục tiêu. Số nòng: 1	
2	Ironclad Blaster	Phạm vi:Súng máy bắn đơn mục tiêu. Số nòng: 1	
3	Vortex Cannon	Phạm vi:Súng máy bắn đơn mục tiêu. Số nòng: 1	

Bảng 4.3. Thông tin cơ bản tháp pháo




2) Thông tin chi tiết

Cấp độ	Sát thương mỗi lần bắn	Tỉ lệ bắn	Phạm vi	Giá xây	Giá bán
1	1250	6 giây	1	200 \$	150 \$
2	2500	6 giây	1.5	400 \$	300 \$
3	3750	6 giây	2.25	600 \$	450 \$

Bảng 4.4. Thông tin chi tiết tháp pháo

4.5.1.3. Tháp Tầm xa(Snipper)

1. Thông tin cơ bản

Cấp độ	Tên	Mô Tả	Hình ảnh
1	Shadow Sentinel	Phạm vi:Súng máy bắn xuyên mục tiêu. Số nòng: 1	
2	Phantom Marksman	Phạm vi:Súng máy bắn xuyên mục tiêu. Số nòng: 1	
3	Eagle Eye Perch	Phạm vi:Súng máy bắn xuyên mục tiêu. Số nòng: 1	

Bảng 4.5. Thông tin cơ bản tháp súng ngắm




2. Thông tin chi tiết

Cấp độ	Sát thương mỗi lần bắn	Tỉ lệ bắn	Phạm vi	Giá xây	Giá bán
1	350	3.5 giây	4.25	300 \$	225 \$
2	700	3.5 giây	5.25	600 \$	450 \$
3	1050	3.5 giây	6.25	900 \$	675 \$

Bảng 4.6. Thông tin chi tiết tháp súng ngắm

4.5.1.4. Tháp Làm chậm(*Time-Wrapper*)

1) Thông tin cơ bản

Cấp độ	Tên	Mô Tả	Hình ảnh
1	Chrono Freezer	Làm chậm tốc độ di chuyển của kẻ địch	
2	Temporal Sentry	Làm chậm tốc độ di chuyển của kẻ địch	
3	Time Warp Tower	Làm chậm tốc độ di chuyển của kẻ địch	

Bảng 4.7. Thông tin cơ bản tháp làm chậm




2) Thông tin chi tiết

Cấp độ	Làm chậm	Phạm vi	Giá xây	Giá bán
1	20%	4.25	200 \$	150 \$
2	40%	5.25	400 \$	300 \$
3	60%	6.25	600 \$	450 \$

Bảng 4.8. Thông tin chi tiết tháp làm chậm

4.5.1.5. Tháp Laser(Laser)

2) Thông tin cơ bản

Cấp độ	Tên	Mô Tả	Hình ảnh
1	Luminous Laser	Laser ghim 1 mục tiêu.	
2	Radiant Blaster	Laser ghim 1 mục tiêu.	
3	Ray Emitter	Laser ghim 1 mục tiêu.	










Bảng 4.9. Thông tin cơ bản tháp laser

2) Thông tin chi tiết

Cấp độ	Sát thương mỗi giây	Phạm vi	Giá xây	Giá bán
1	200	1	200 \$	150 \$
2	400	1.5	400 \$	300 \$
3	600	2.25	600 \$	450 \$




Bảng 4.10. Thông tin chi tiết tháp laser

4.5.2. Quân địch

Tên	Hình ảnh	Tốc độ	Máu	Tiền hạ gục	Sát thương lên căn cứ
G solider		0.75m/s	40	3 \$	1 máu
Aka Solider		0.75m/s	160	5 \$	1 máu
Car		1.7m/s	280	7 \$	1 máu
Truck		1m/s	1500	10 \$	1 máu
Tank Raw		1m/s	5500	20 \$	1 máu
Brown Steal		1m/s	10000	30 \$	1 máu
Black Steal		1m/s	15000	40 \$	1 máu
Tank Double		1m/s	20000	50 \$	2 máu
Tank Radar		1m/s	50000	60 \$	3 máu



Bảng 4.11. Thông tin chi tiết kẻ địch

4.5.3. Chương ngại vật trong game

Loại	Máu	Tiền hạ gục	Không gian chiếm dụng
	1600	10 \$	1 ô
	5000	30 \$	1 ô
	1600	10 \$	1 ô

Bảng 4.12. Thông tin chi tiết chương ngại vật

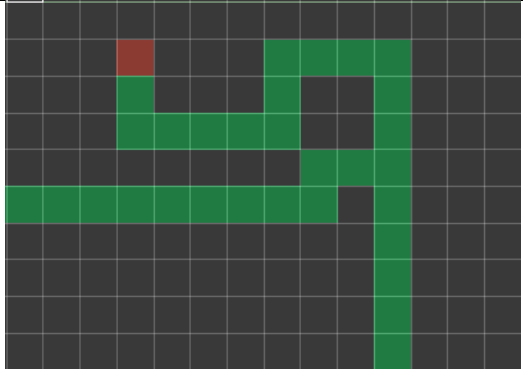
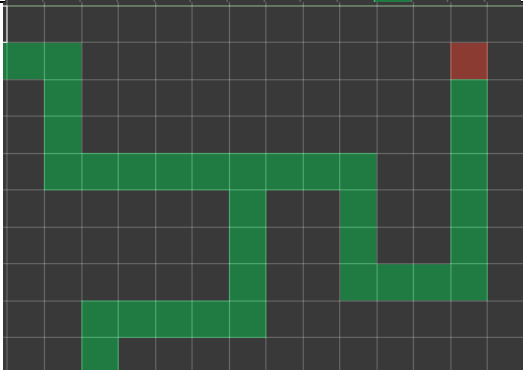
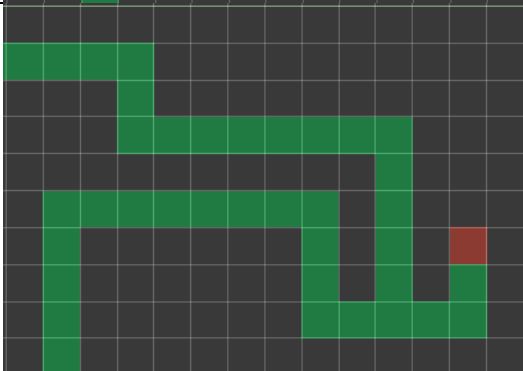
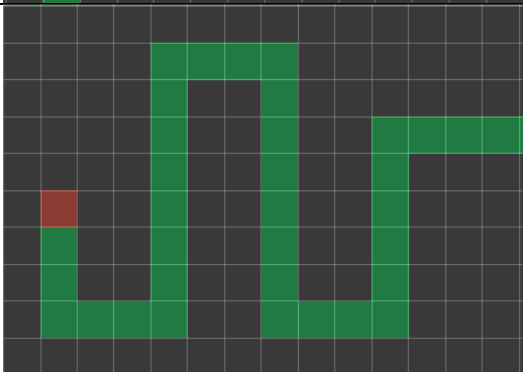
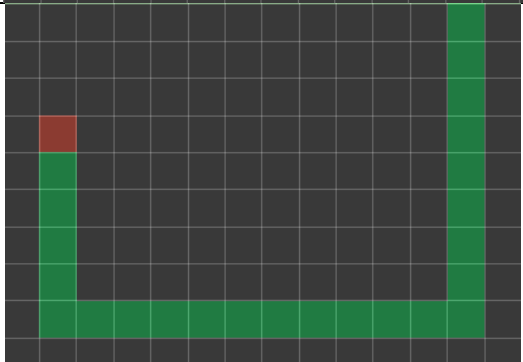
4.5.4. Hỗ trợ trong game

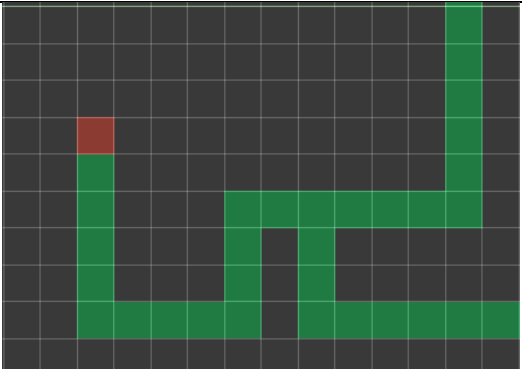
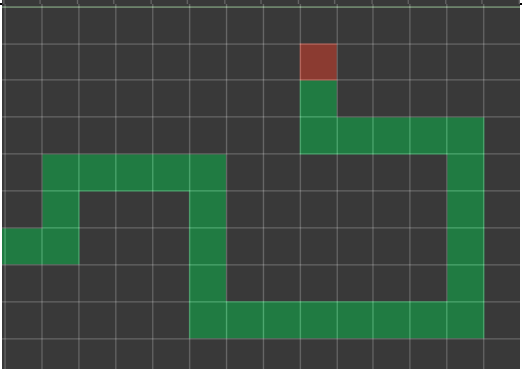
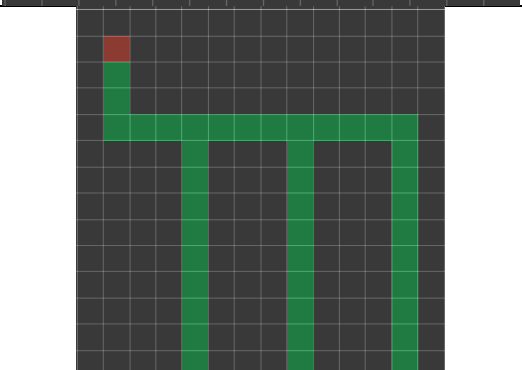
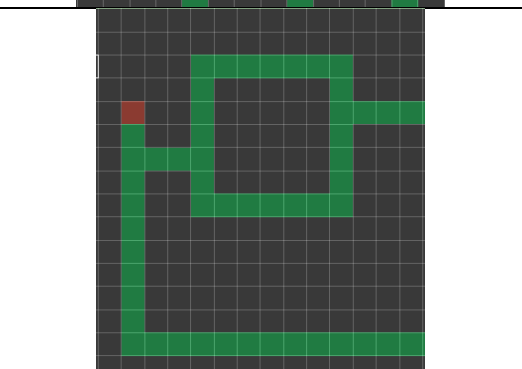
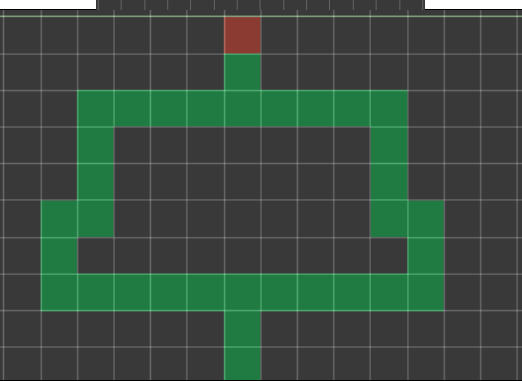
Tên	Hình ảnh	Chức năng	Thời gian hồi
Dịch chuyển		Dịch chuyển kẻ địch về vị trí ban đầu	5 giây
Mìn		Phát nổ gây sát thương lên tất cả kẻ địch ở trong phạm vi	5 giây

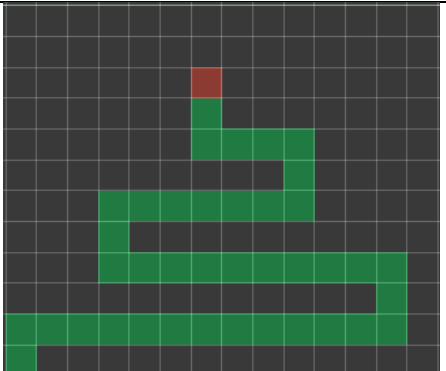
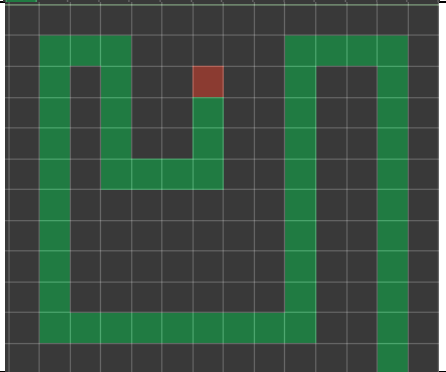
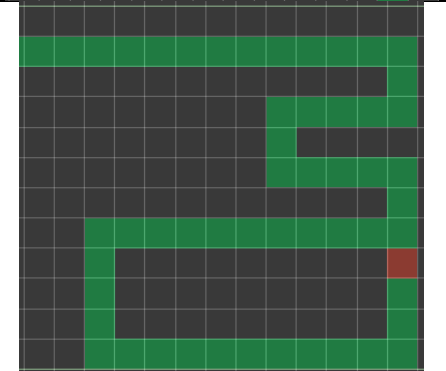
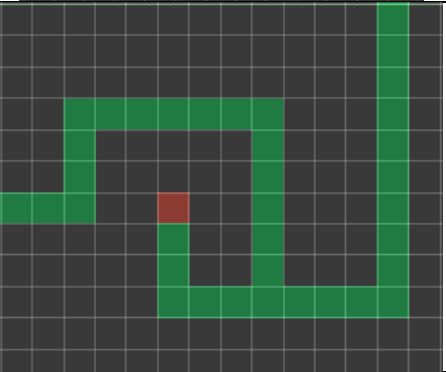
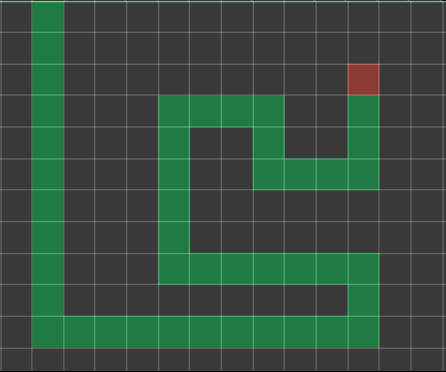
Bảng 4.13. Thông tin chi tiết hỗ trợ

4.5.5. Thiết kế level trong game

Level	Kích thước(đơn vị ô)	Tiền sau mỗi level	Đường đi của quân địch
1	14 x 10	100	
2	14 x 10	200	
3	14 x 10	300	
4	14 x 10	400	

5	14 x 10	500	
6	14 x 10	600	
7	14 x 10	700	
8	14 x 10	800	
9	14 x 10	900	

10	14 x 10	1000	
11	14 x 10	1100	
12	14 x 14	1200	
13	14 x 16	1300	
14	14 x 10	1400	

15	14 x 12	1500	
16	14 x 12	1600	
17	14 x 12	1700	
18	14 x 12	1800	
19	14 x 12	1900	

CHƯƠNG 5. THIẾT KẾ HỆ THỐNG

Trong chương trước đã giới thiệu về bối cảnh, cốt chuyện, cách chơi, thông số chi tiết của kẻ địch, màn chơi... Chương này sẽ thiết kế và triển khai hệ thống

5.1. Thiết kế kiến trúc

Kiến trúc OOP (Object-Oriented Programming) là một mô hình lập trình dựa trên khái niệm "đối tượng" để thiết kế và xây dựng các ứng dụng phần mềm. Trong OOP, các đối tượng được xem như là các thực thể, mỗi thực thể có các thuộc tính và phương thức. [6]

5.1.1. Đối tượng (Object)

- Là một thực thể trong thế giới thực hoặc trong phần mềm, có thể đại diện cho một người, một vật, hoặc một khái niệm.
- Đối tượng có hai đặc tính chính: thông tin và hành vi.
- Thông tin được lưu trữ trong các thuộc tính (properties hoặc fields).
- Hành vi được thể hiện qua các phương thức (methods hoặc functions).

5.1.2. Lớp (Class)

Lớp một bản thiết kế hoặc khuôn mẫu để tạo ra các đối tượng.

Một lớp định nghĩa các thuộc tính và phương thức mà các đối tượng thuộc lớp đó sẽ có.

5.1.3. Đặc tính cơ bản của OOP

1. Tính đóng gói (Encapsulation):

Là việc gói gọn dữ liệu và các phương thức thao tác dữ liệu trong một đối tượng.

Chỉ cho phép truy cập dữ liệu thông qua các phương thức công khai (public methods) để bảo vệ dữ liệu bên trong.

2. Tính kế thừa (Inheritance):

Cho phép một lớp con kế thừa các thuộc tính và phương thức từ một lớp cha.

Giúp tái sử dụng mã nguồn và dễ dàng mở rộng các tính năng.

3. Tính đa hình (Polymorphism):

Cho phép các đối tượng thuộc các lớp khác nhau có thể được xử lý thông qua cùng một giao diện chung.

Gồm hai loại: đa hình tĩnh (static polymorphism) thông qua quá tải phương thức (method overloading) và đa hình động (dynamic polymorphism) thông qua ghi đè phương thức (method overriding).

4. Tính trừu tượng (Abstraction):

Là quá trình đơn giản hóa các hệ thống phức tạp bằng cách chỉ tập trung vào những chi tiết cần thiết và che giấu đi những chi tiết phức tạp không cần thiết.

Được thực hiện thông qua các lớp trừu tượng (abstract classes) và giao diện (interfaces). [7]

5.2. Thiết kế giao diện

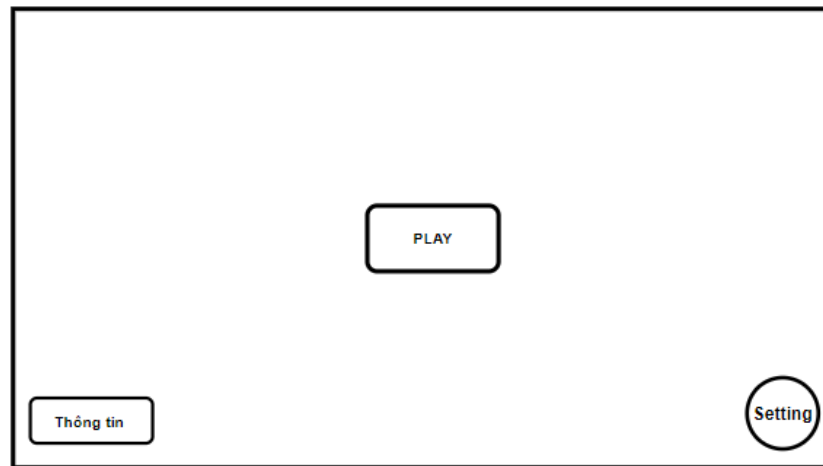
5.2.1. Thiết kế màn hình

Thông Tin	Thông số
Kích thước màn hình	Tiêu chuẩn: 720 x 1280
Chế độ màn hình	Toàn màn hình
Front chữ	SVN-Neutraface 2

Bảng 5.1. Thông tin màn hình

Phần thiết kế màn hình được dựng bằng phần Draw.io, sử dụng một số hình vẽ đơn giản.

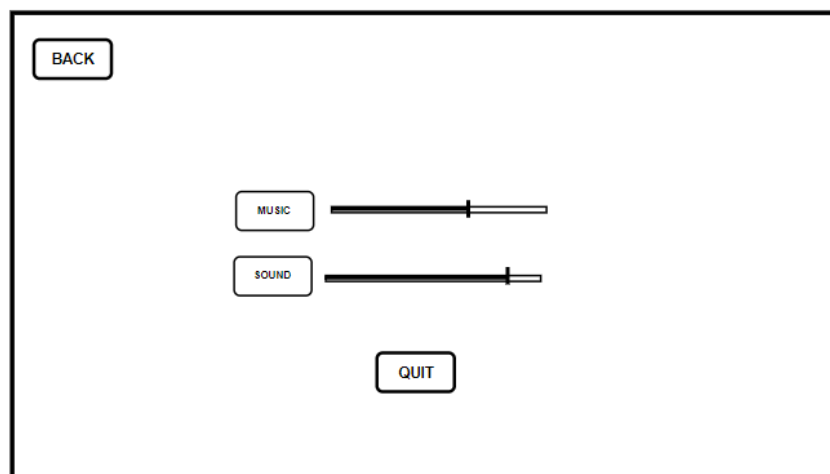
5.2.2. Màn hình Home



Hình 5.1. Màn hình thiết kế home

Đây là màn hình khi người chơi bắt đầu vào game, màn hình bao gồm 3 nút Play, Thông tin nhà phát triển và nút cài đặt.

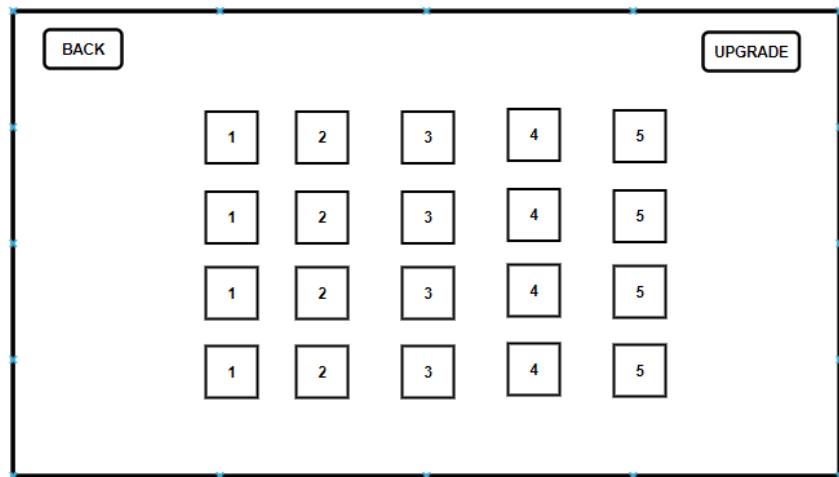
5.2.3. Màn hình cài đặt



Hình 5.2. Màn hình thiết kế cài đặt

Màn hình này xuất hiện khi người chơi ấn nút setting tại home.

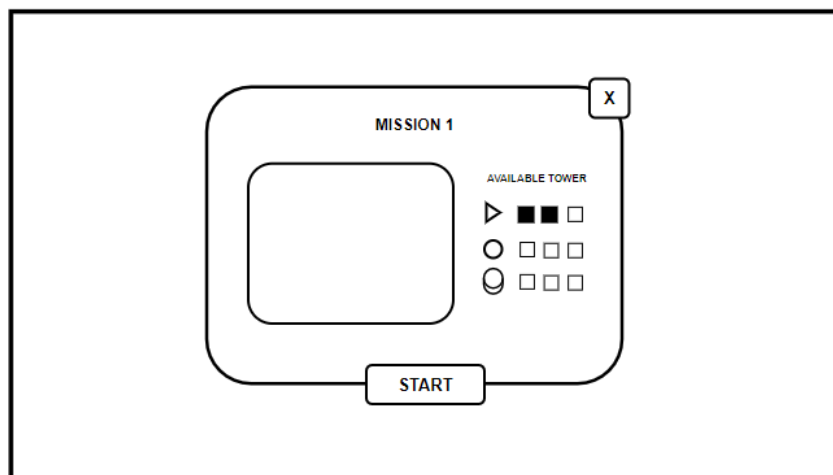
5.2.4. Màn hình chọn Level



Hình 5.3. Màn hình thiết kế chọn level

Đây là màn hình xuất hiện khi người chơi click nút play tại màn hình home. Màn hình này bao gồm danh sách những level đã mở khóa và chưa mở khóa. Người chơi có thể chọn một level để chơi. Nút Upgrade để vào kho nâng cấp tháp.

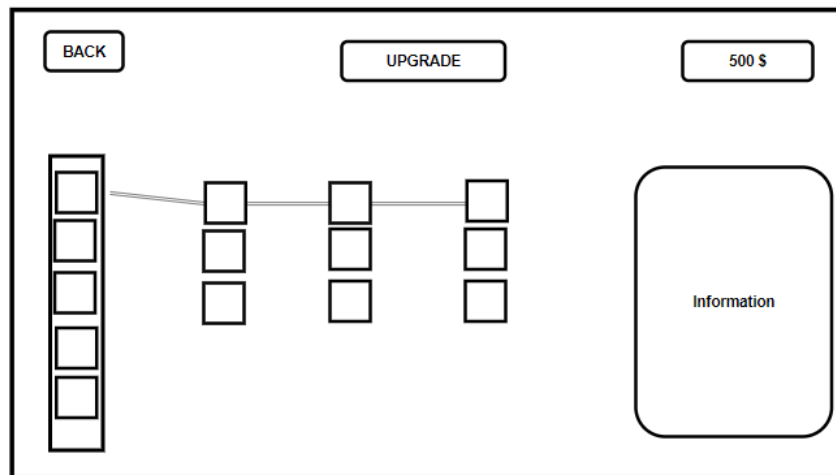
5.2.5. Màn hình bắt đầu Level



Hình 5.4. Màn hình thiết kế bắt đầu level

Màn hình này xuất hiện khi người chơi chọn một level, trong đây sẽ bao gồm thông tin về level và những giới hạn level và loại tháp có thể xây dựng và một bức ảnh demo bản đồ của nhiệm vụ đó.

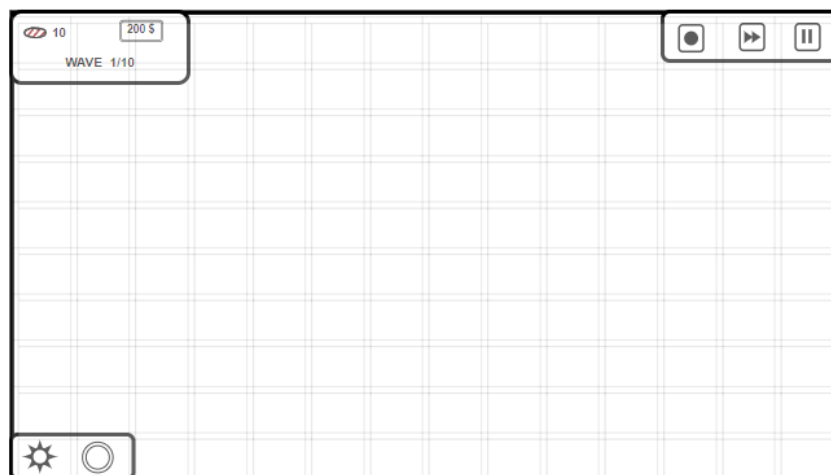
5.2.6. Màn hình Upgrade



Hình 5.5. Màn hình thiết kế upgrade

Màn hình này xuất hiện khi người chơi click nút Upgrade tại màn hình chọn level. Trong màn hình này sẽ bao gồm những tháp đã được mở khóa và chưa được mở khóa người chơi sẽ chọn vào tháp muốn nâng cấp để xem thông tin. Thông tin sẽ hiển thị trên bảng góc dưới bên phải màn hình.

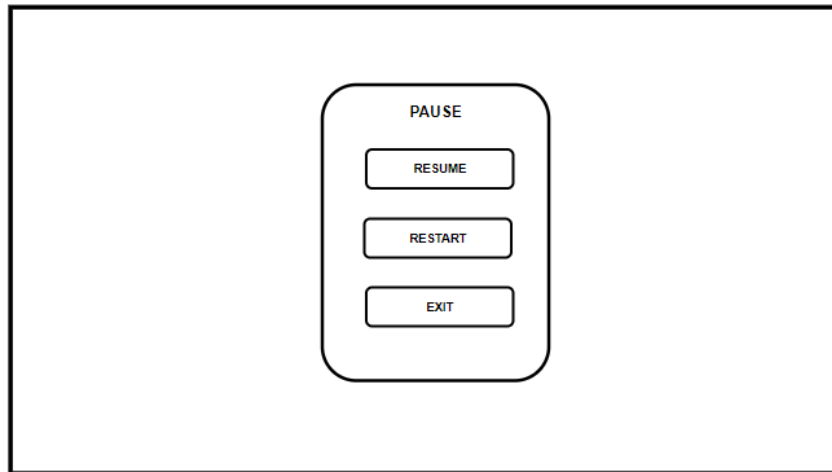
5.2.7. Màn hình trong game



Hình 5.6. Màn hình thiết kế trong game

Đây là màn hình trong game. Bao gồm thông tin wave, tiền, máu. Góc dưới bên trái hiển thị thông tin ability có thể sử dụng.

5.2.8. Màn hình tạm dừng game



Hình 5.7. Màn hình thiết kế tạm dừng game

Màn hình này xuất hiện khi người chơi click nút tạm dừng trong khi đang chơi.

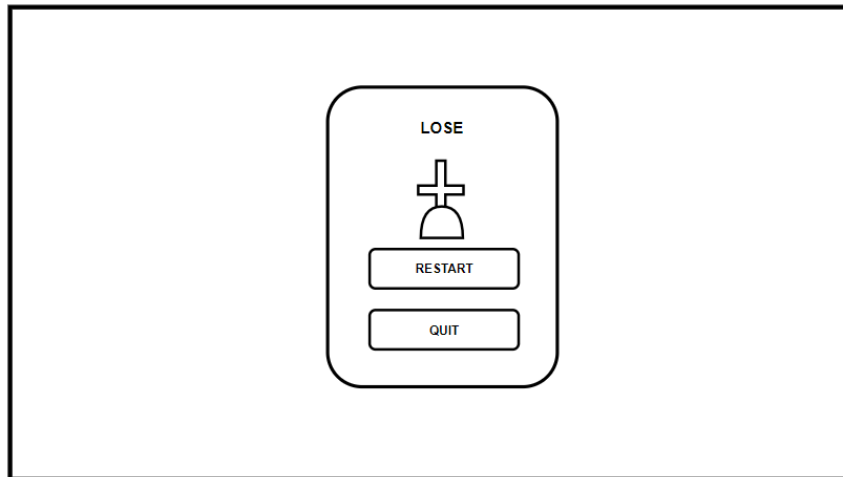
5.2.9. Màn hình chiến thắng



Hình 5.8. Màn hình thiết kế chiến thắng

Màn hình này xuất hiện khi người chơi hoàn thành nhiệm vụ. Bao gồm thông tin phần thưởng nhận được và số sao nhận được.

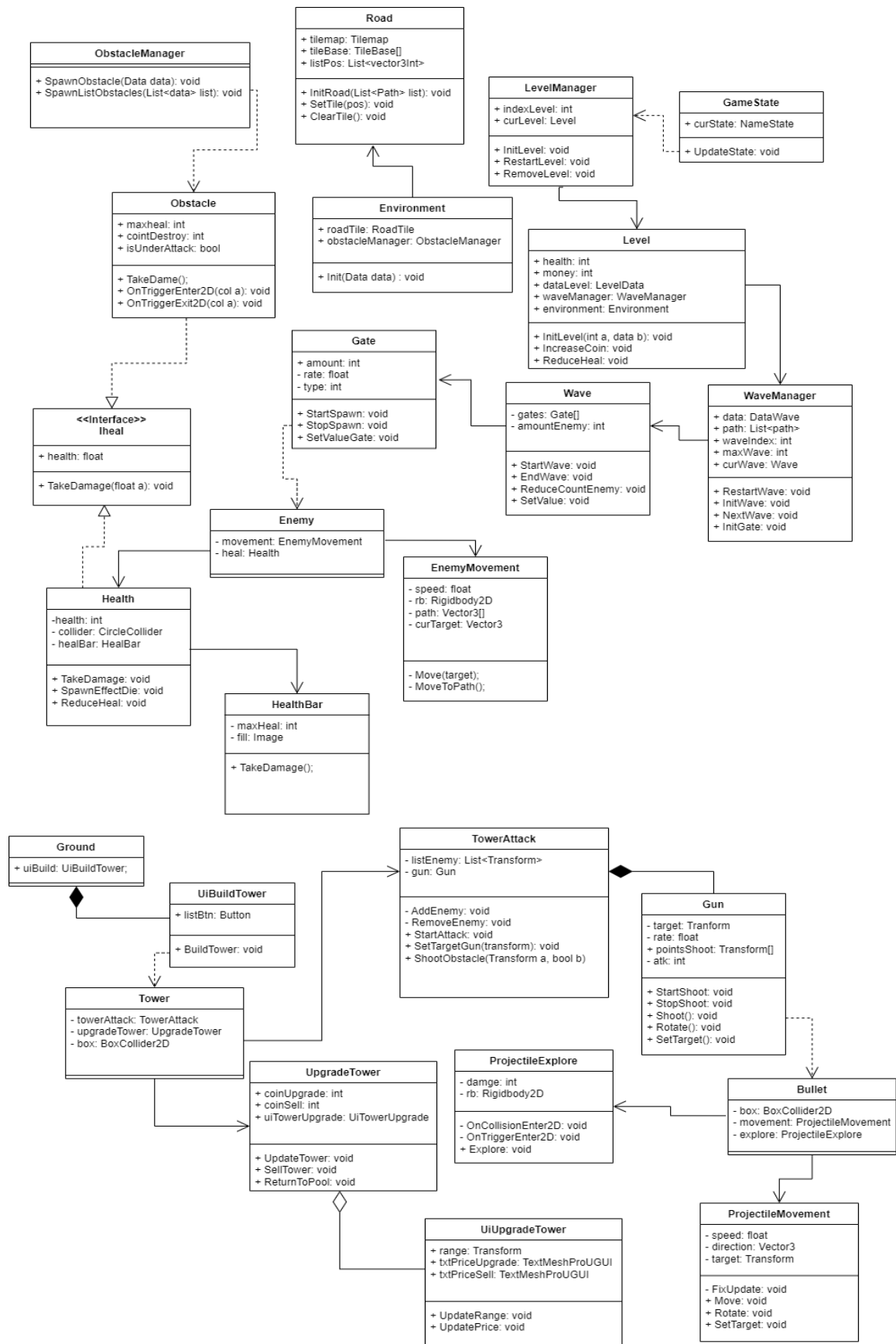
5.2.10. Màn hình thua cuộc



Hình 5.9. Màn hình thiết kế thua cuộc

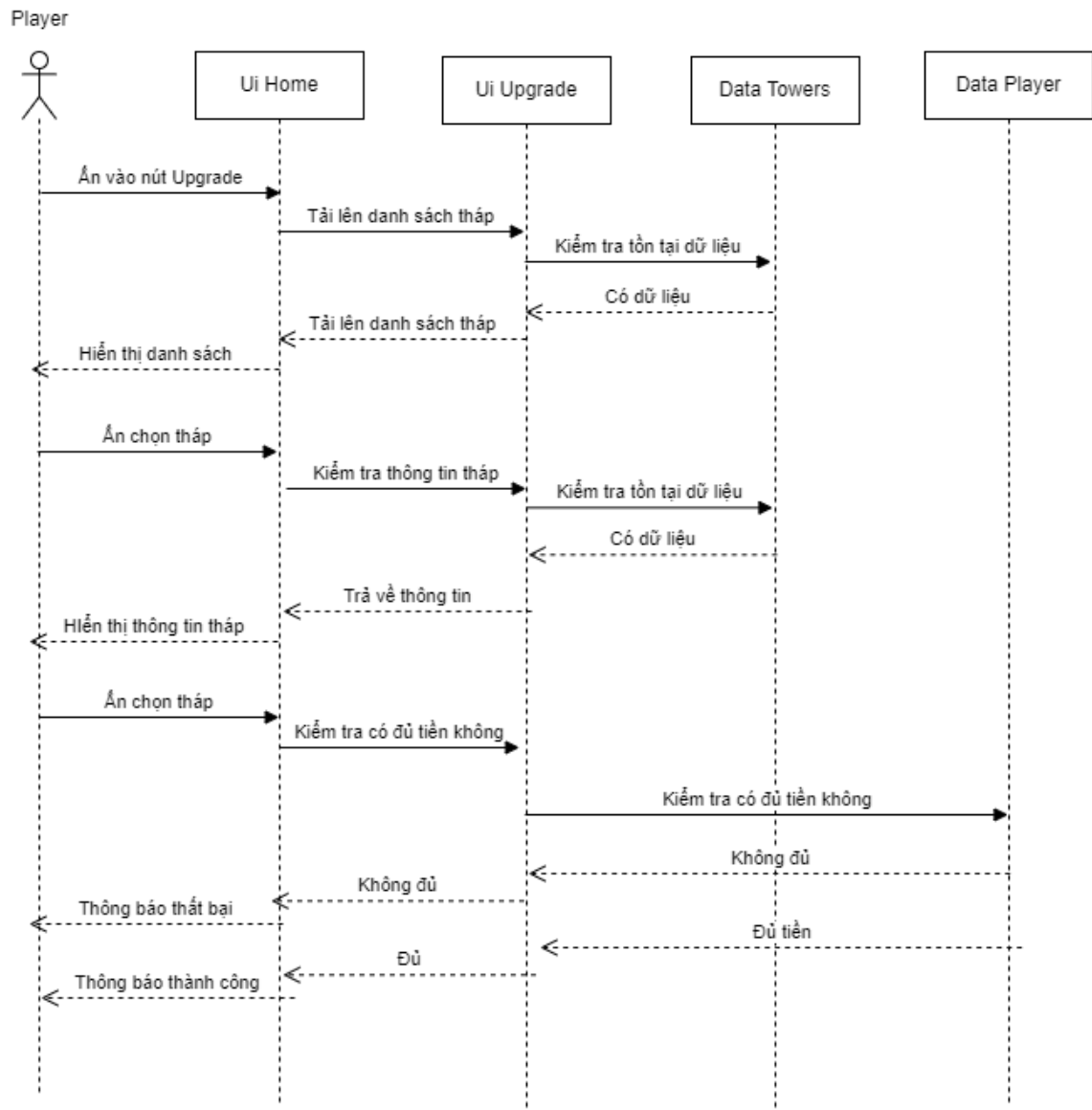
Màn hình này xuất hiện khi người chơi phòng thủ thất bại.

5.3. Thiết kế lớp



Hình 5.10. Biểu đồ lớp

Biểu đồ trình tự nâng cấp tháp

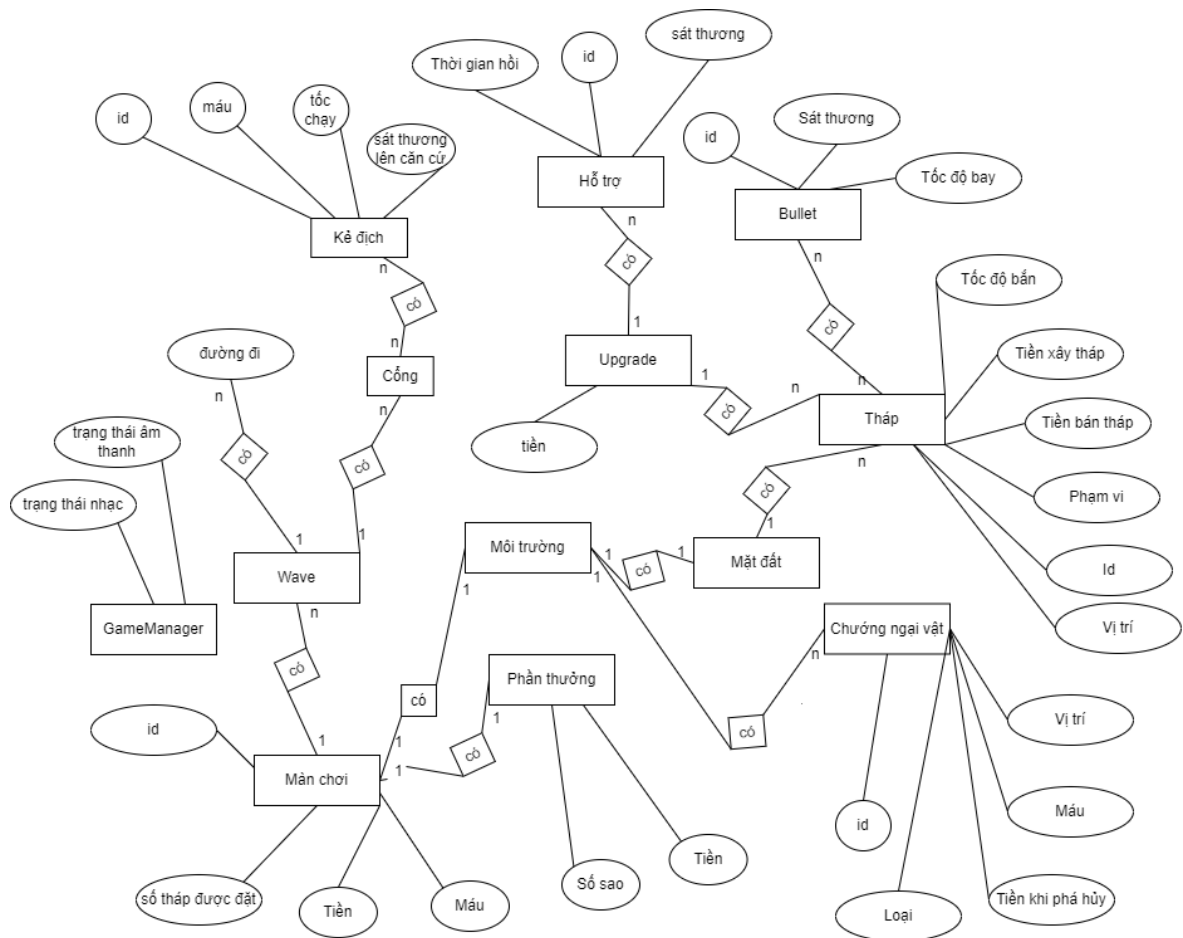


Hình 5.11. Biểu đồ trình tự nâng cấp tháp

Trong phần này sẽ trình bày biểu đồ lớp gameplay chính trong game và biểu đồ trình tự nâng cấp tháp trong game. [8]

5.4. Thiết kế cơ sở dữ liệu

Mô hình thiết kế trong game này sẽ không sử dụng cơ sở dữ liệu mà sẽ lưu trữ dữ liệu theo dạng File. Cách lưu dạng file được sử dụng là ScriptableObjects hay lưu trực tiếp tại đối tượng trên Scene hoặc lưu bằng PlayerPrefs vì thế trong biểu đồ thực thể liên kết sẽ có một vài điểm khác biệt so với biểu đồ có sử dụng cơ sở dữ liệu. [8]



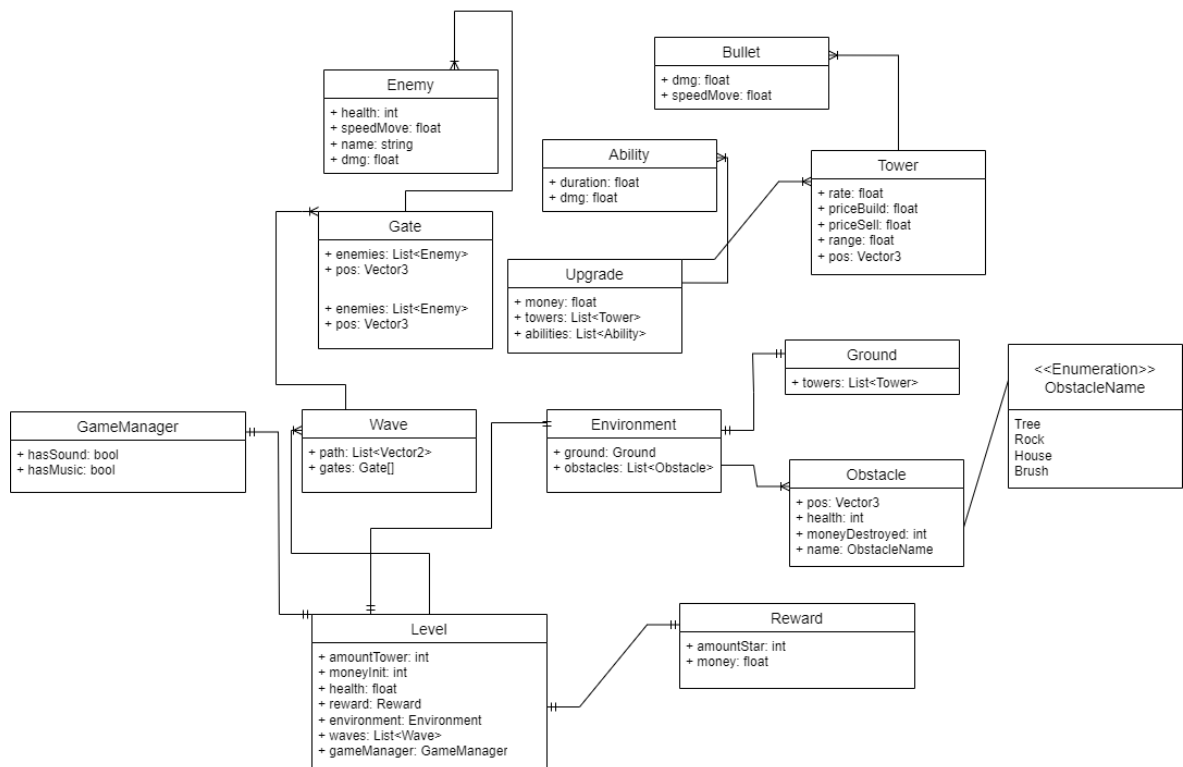
Hình 5.12. Biểu đồ cơ sở dữ liệu

Giải thích sơ đồ thực thể liên kết:

- Tháp: có các thuộc tính như tốc độ bắn đạn, số tiền để xây tháp, số tiền nhận được khi bán tháp, phạm vi tấn công của tháp, id, vị trí đặt tháp.
- Đạn: sát thương gây ra, tốc độ di chuyển, id

- Môi trường: mặt đất, chướng ngại vật.
- Upgrade: tháp, hỗ trợ
- Wave: kẻ địch, đường đi của kẻ địch.
- Màn chơi: wave, phần thưởng, môi trường, số tháp có thể đặt, id, máu căn cứ, tiền ban đầu.
- Kẻ địch: id, máu, sát thương khi vào căn cứ, tốc độ chạy.
- Chướng ngại vật: vị trí, máu, id, loại
- Loại chướng ngại vật: bụi rậm, cây, đá, nhà.

Các lớp lưu trữ:



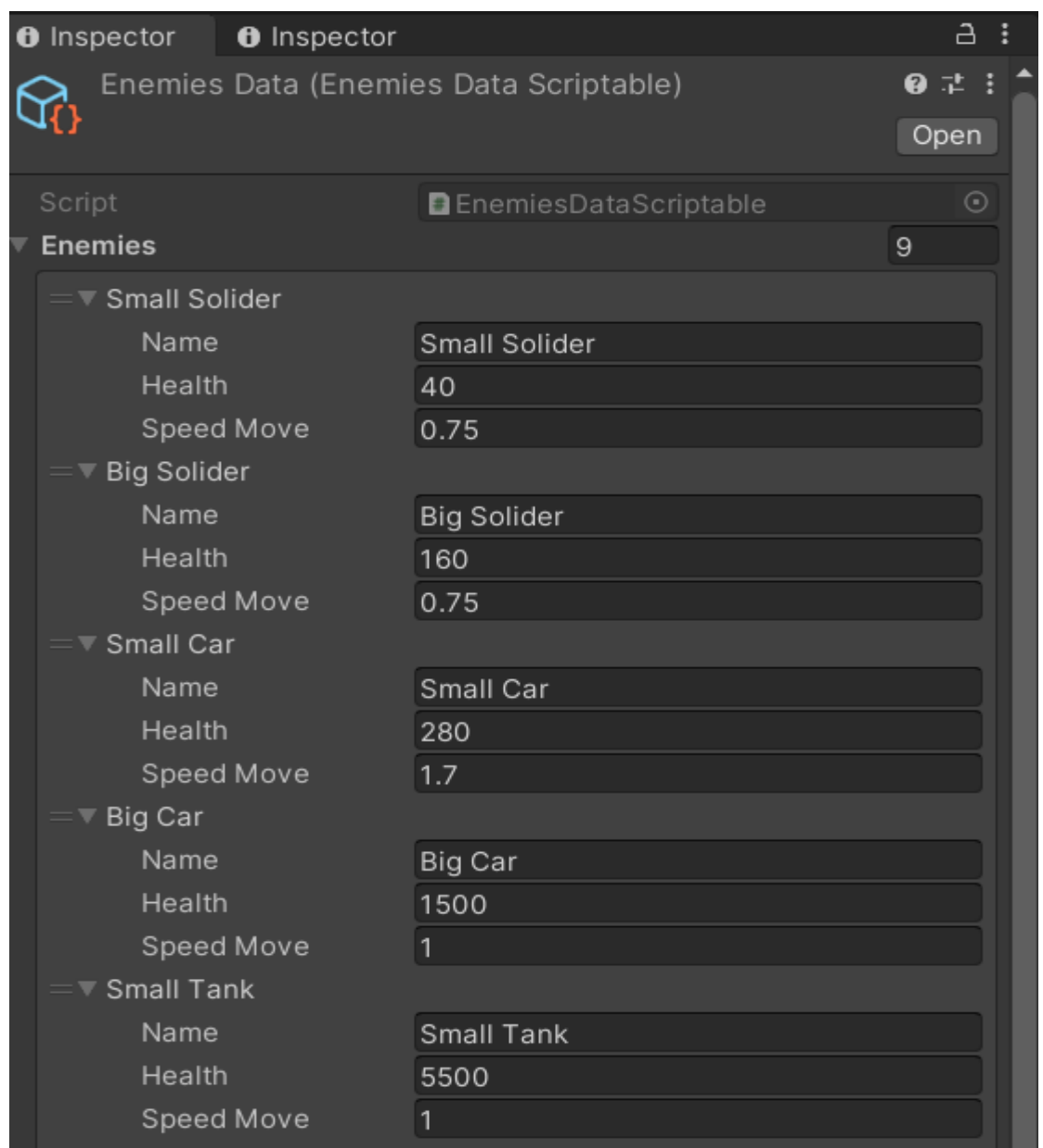
Hình 5.13. Biểu đồ lớp lưu trữ

Trong Unity đối với game offline có 3 cách phổ biến để lưu trữ dữ liệu:

- *ScriptableObjects*: đây là một class của unity được sử dụng để lưu trữ những dữ liệu không thay đổi giữa các lần chơi. Những file scriptableObject sẽ được lưu dưới dạng file và tải lên khi trò chơi bắt đầu. Dữ liệu lưu trong scriptableObject sẽ không bị xóa khi người dùng xóa dữ liệu game.

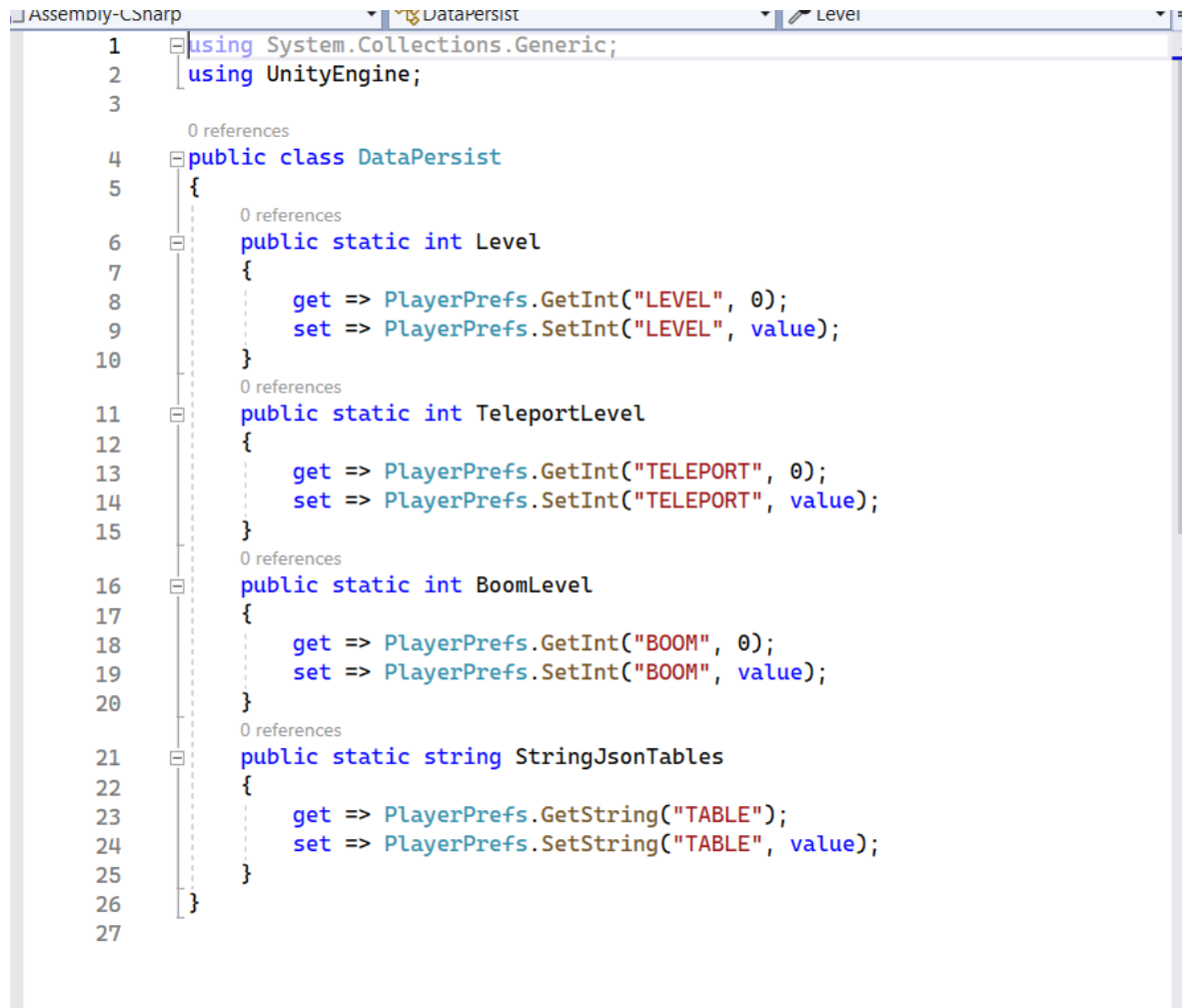
- *PlayerPrefs*: Class này thuộc unity nó lưu trữ dưới dạng cặp key – value và thường được sử dụng để lưu những dữ liệu đơn giản. Dữ liệu khi lưu trong PlayerPrefs sẽ bị xóa khi người dùng xóa dữ liệu trò chơi.
- *Lưu trên Object*: kiểu lưu này là dữ liệu sẽ được lưu trực tiếp trên các đối tượng trên scene thông qua inspector, khi người chơi vào lại game thì dữ liệu sẽ reset về giá trị ban đầu.

Ảnh minh họa:



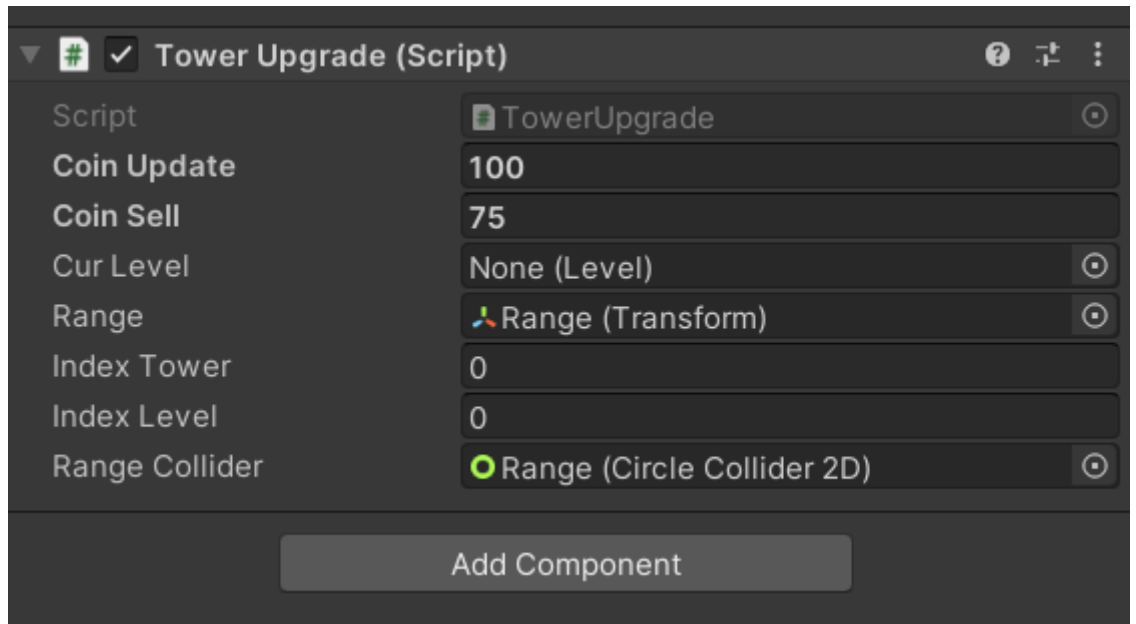
Hình 5.14. Scriptable Object

Những giá trị như máu, tốc chạy và tên của kẻ địch được lưu trữ dưới dạng scriptable Object, những dữ liệu này là dữ liệu tĩnh.



Hình 5.15. Minh họa Playerprefs

Hình trên mô tả cách lưu dữ liệu bằng PlayerPrefs, những giá trị như level, levelTeleport, BoomLevel là những dữ liệu sẽ thay đổi khi người chơi nâng cấp hoặc mở khóa.



Hình 5.16. Minh họa lưu dữ liệu trên scene

Hình trên là ví dụ về lưu trữ trên scene, những giá trị như Coin Update, Coin Sell, IndexTower, Index Level sẽ được reset mỗi khi người chơi vào lại trò chơi.

5.5. Công cụ sử dụng

Trong dự án có sử dụng một số công cụ như:

- IDE(Visual studio): đây là công cụ không chỉ để phát triển trò chơi mà còn được sử dụng phổ biến trong những hệ thống như website, application... Nó được tích hợp rất nhiều công cụ và tính năng cần thiết để phát triển phần mềm như: trình chỉnh sửa, trình biên dịch, trình gỡ lỗi... và rất nhiều tính năng hỗ trợ khác.
- Engine(Unity): một game engine là một phần mềm hoặc tập hợp các công cụ được thiết kế để giúp các nhà phát triển xây dựng và triển khai các trò chơi một cách hiệu quả. Game engine cung cấp các chức năng cần thiết để phát triển một trò chơi, bao gồm đồ họa, âm thanh, vật lý, trí tuệ nhân tạo, và các công cụ quản lý tài nguyên.
- Photoshop: trong quá trình phát triển trò chơi đặc biệt là trò chơi 2D sẽ cần rất nhiều hình ảnh và những gói đồ họa, sẽ có những hình ảnh không

thật sự phù hợp cho style của game hoặc thiếu một số hình ảnh. Photoshop sẽ giải quyết được vấn đề trên.

- Draw.io: đây là phần mềm dùng để chỉnh sửa và tạo các sơ đồ trong quá trình phân tích thiết kế hệ thống như: sơ đồ usecase, biểu đồ lớp, biểu đồ trình tự...

5.6. Kết quả đạt được

Sau 2 tháng tìm hiểu và phát triển, một phiên bản đầu tiên đã được hoàn thiện đây là phiên bản quan trọng nhất triển khai tất cả những tính năng chính của trò chơi những hiệu ứng dữ liệu nền tảng để làm cơ sở phát triển thêm những phiên bản về sau.

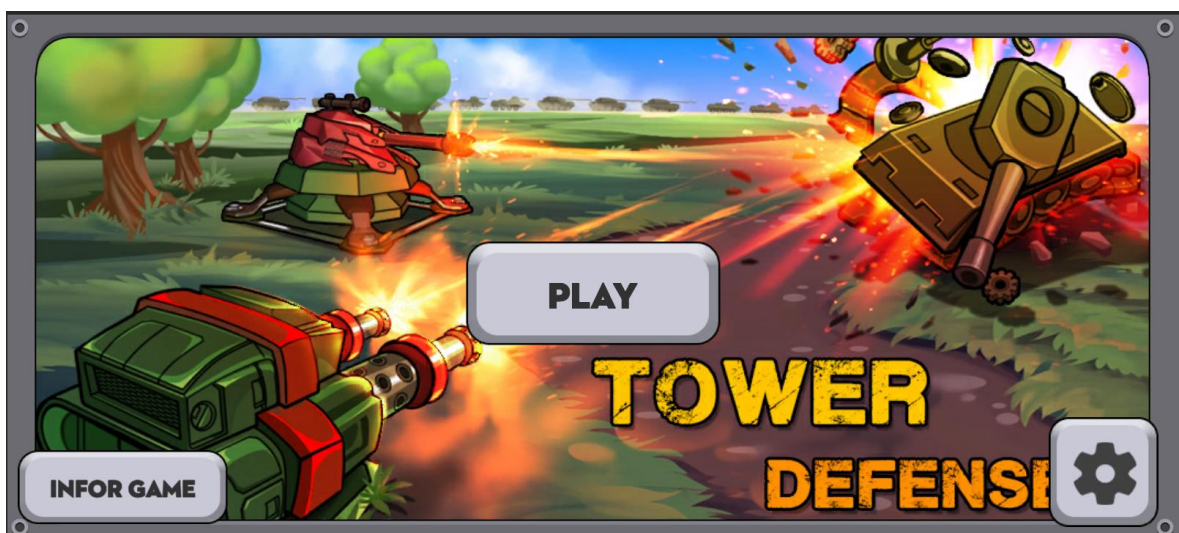
Thông tin về dự án:

Thông tin	Chi tiết
Dung lượng toàn bộ dự án	3.15 GB
Dung lượng sản phẩm	30 MB

Bảng 5.2. Thông tin dung lượng dự án

5.7. Màn hình chức năng chính

5.7.1. Màn hình Home



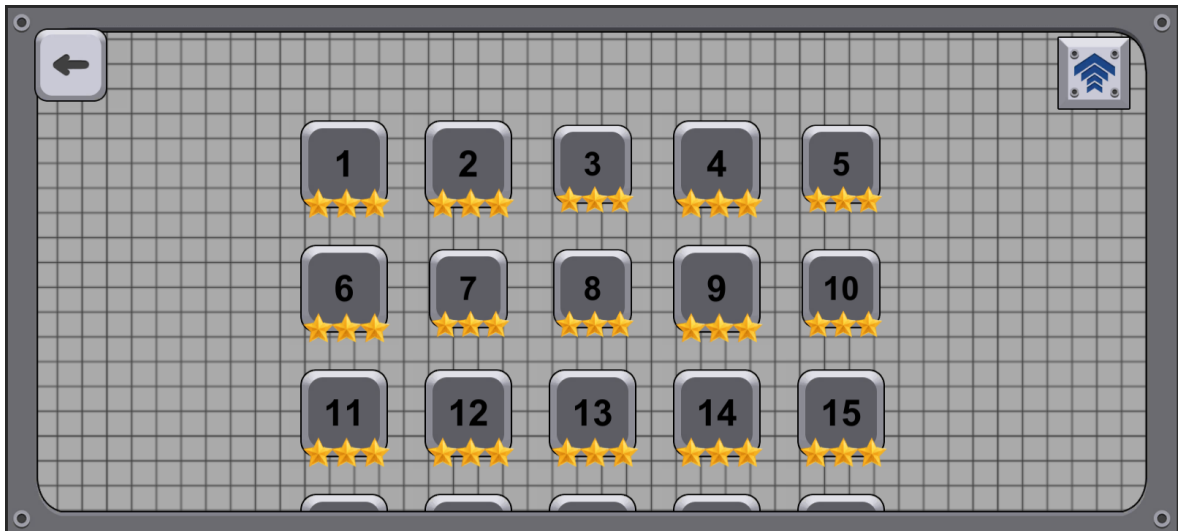
Hình 5.17. Màn hình home

Đây là màn hình đầu tiên khi người chơi bắt đầu vào game, màn hình này bao gồm những nút sau: Cài đặt, Play và InforGame.

Như đã trình bày ở phần thiết kế:

- Click Play sẽ mở ra danh sách level để người chơi có thể chọn
- Click Cài đặt để tùy chỉnh âm thanh trong game
- Click InforGame để xem thông tin về Game

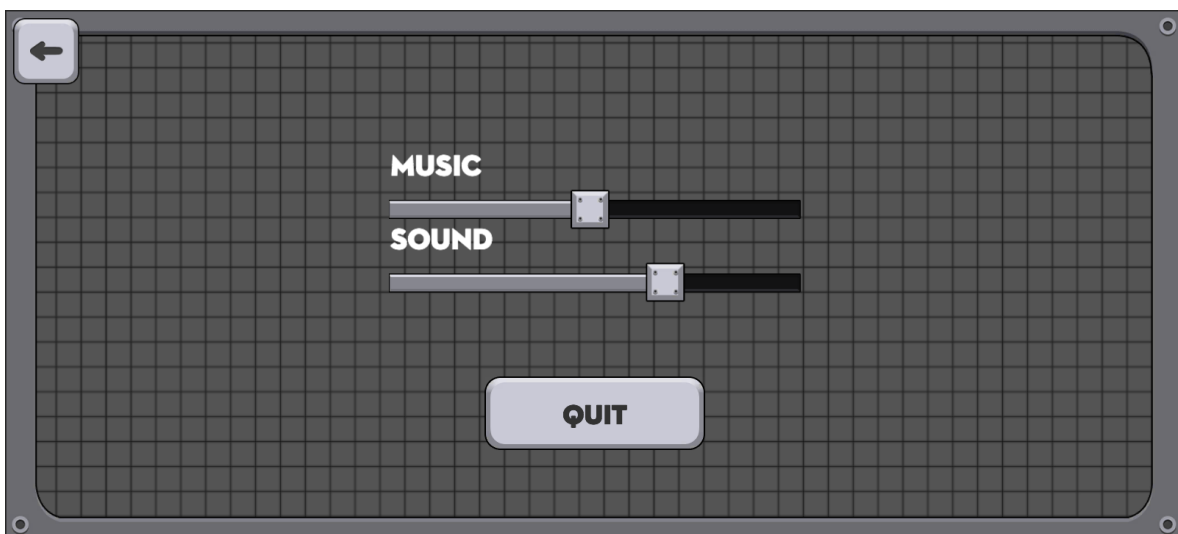
5.7.2. Màn hình chọn màn chơi



Hình 5.18. Màn hình chọn màn chơi

Màn hình này sẽ hiển thị màn chơi người chơi đã chinh phục và số sao đạt được ở mỗi màn.

5.7.3. Màn hình cài đặt



Hình 5.19. Màn hình cài đặt

Màn hình này cho phép người chơi thay đổi âm lượng của game.

5.7.4. Màn hình thông tin game



Hình 5.20. Màn hình thông tin game

5.7.5. Màn hình nâng cấp tháp



Hình 5.21. Màn hình nâng cấp

Trong màn hình nâng cấp tháp sẽ bao gồm một thanh menu cho phép người chơi chọn tháp cần nâng cấp. Một bảng thông tin hiển thị thông tin chi tiết của từng tháp và số tiền hiện tại của người chơi.

5.7.6. Màn hình map thung lũng



Hình 5.22. Màn hình map thung lũng

5.7.7. Màn hình map sa mạc



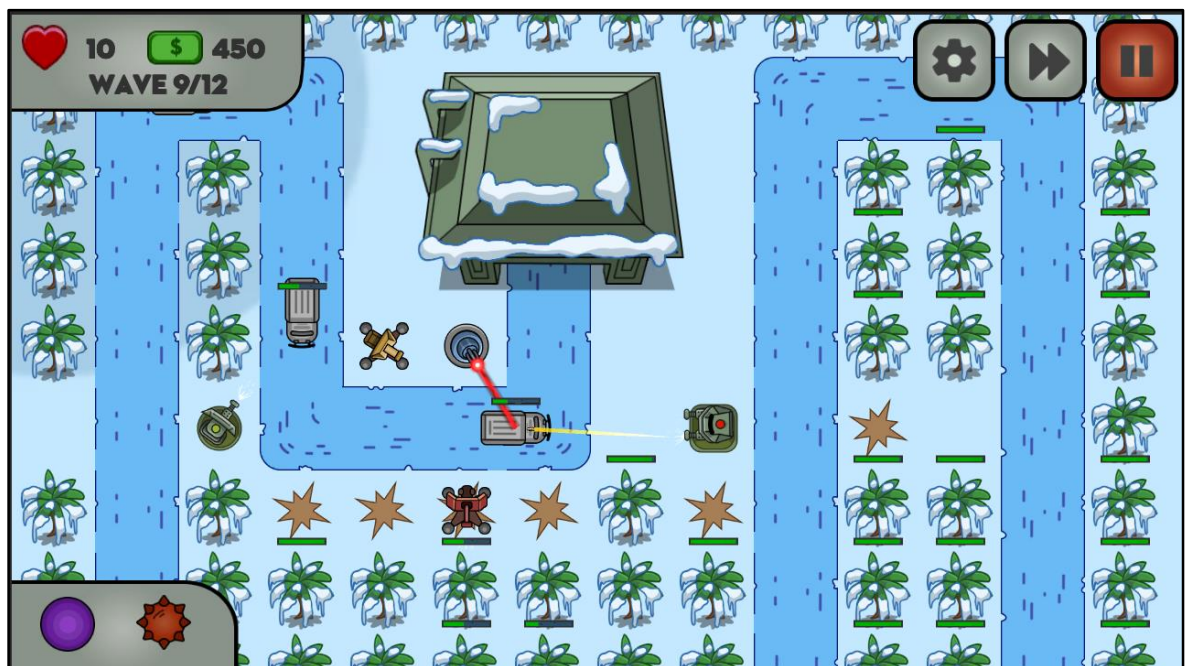
Hình 5.23. Màn hình map sa mạc

5.7.8. Màn hình map đường phố



Hình 5.24. Màn hình map đường phố

5.7.9. Màn hình map bắc cực



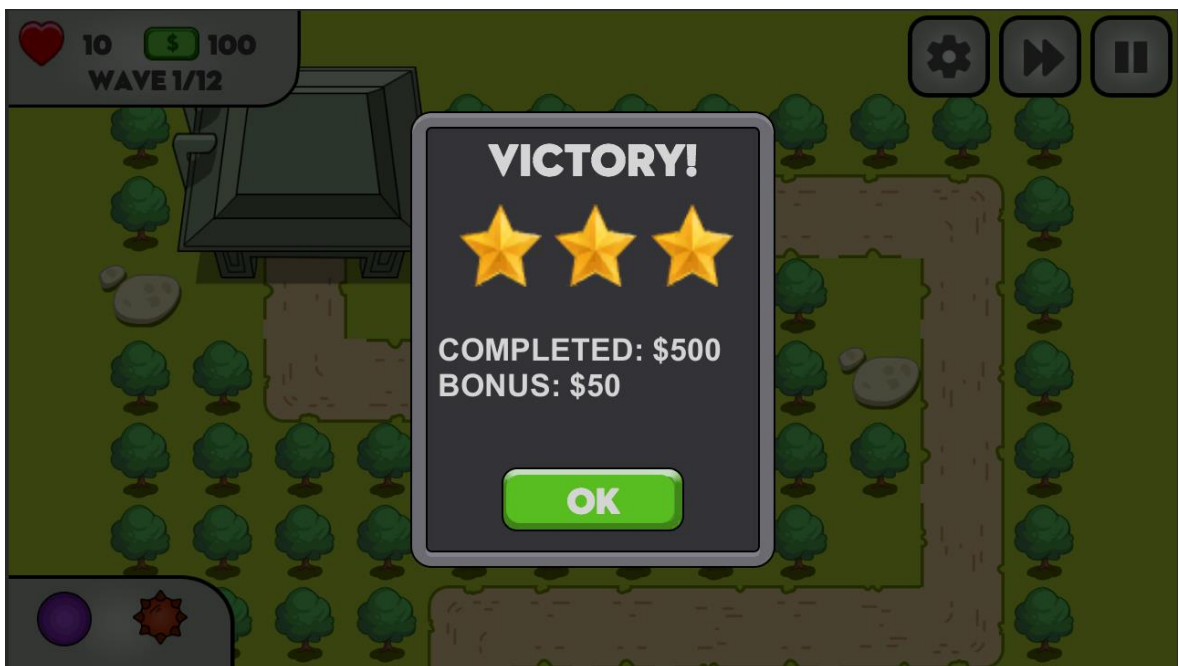
Hình 5.25. Màn hình map bắc cực

5.7.10. Màn hình thua cuộc



Hình 5.26. Màn hình thua cuộc

5.7.11. Màn hình chiến thắng



Hình 5.27. Màn hình chiến thắng

CHƯƠNG 6. KIỂM THỬ PHẦN MỀM

Sau khi thiết kế hoàn thiện ở chương trước, trong chương này sẽ kiểm thử phần mềm tìm ra những lỗi và vấn đề cần khắc phục.

6.1. Kế hoạch kiểm thử

Mục đích:

Tài liệu kế hoạch kiểm thử này đưa ra các mục đích sau:

- Kiểm thử thông số chi tiết của kế hoạch và tháp
- Kiểm thử màn hình

Phạm vi:

Tài liệu kế hoạch kiểm thử này được áp dụng cho việc kiểm thử chức năng giao diện và thông số chi tiết.

- **Phần cứng: pc**

Máy tính cá nhân

CPU	RAM	SSD	Architecture
Intel(R) Core(TM) i7-6820HQ CPU @ 2.70GHz 2.70 GHz	8 GB	237 GB	64 bit

Bảng 6.1. Thông tin máy tính kiểm thử

Điện thoại cá nhân

CPU	RAM	HỆ ĐIỀU HÀNH	SSD
Snapdragon 665 AIE	4 GB	Android 9 (Pie)	64 GB

Bảng 6.2. Thông tin điện thoại kiểm thử

6.2. Kiểm thử chức năng

Kiểm thử những chức năng như đặt tháp, đặt hỗ trợ, nâng cấp tháp, nâng cấp hỗ trợ, tạm dừng game.

STT	Tên chức năng	Quy trình thực hiện	Kết quả kỳ vọng	Kết quả
1	Đặt tháp	Chọn một vùng đất trống trên bản đồ. Click vào đó. Chọn tháp muốn đặt.	Khi ấn chọn vùng trống trên mặt đất sẽ hiển thị một popup chứa danh sách những tháp có thể đặt. Sau khi click vào tháp muốn đặt popup ẩn đi và tháp được xây dựng.	Đạt
2	Đặt hỗ trợ	Click vào đường đi của kẻ địch, ấn chọn loại hỗ trợ muốn sử dụng.	Khi ấn vào đường đi của địch một popup bao gồm những loại hỗ trợ hiện lên. Khi chọn một loại hỗ trợ thì popup đóng lại và hỗ trợ xuất hiện trên đường đi của kẻ địch.	Đạt
3	Nâng cấp tháp	Vào cửa hàng nâng cấp chọn một tháp muốn nâng cấp, ấn mua tháp.	Khi ấn mua thông tin tháp sẽ được cập nhật và trừ tiền.	Đạt
4	Nâng cấp hỗ trợ	Vào cửa hàng nâng	Sau khi ấn mua hỗ trợ,	Đạt

		cấp chọn một loại hỗ trợ muốn nâng cấp, ấn mua để nâng cấp.	thông tin được thay đổi và trừ tiền. Cập nhật thay đổi khi vào game.	
5	Tạm dừng game	Ấn nút Pause khi chơi game. Ấn lại lần nữa để tiếp tục.	Khi ấn nút tạm dừng kẻ địch và tháp tạm dừng. Khi ấn lại lần 2 kẻ địch và tháp tiếp tục hoạt động.	Đạt

Bảng 6.3. Kiểm thử chức năng

6.3. Kiểm thử sức mạnh từng loại tháp trong game

STT	Tên tháp	Quy trình thực hiện	Kết quả kỳ vọng	Kết quả
1	Machine Gun Level: 1 Kẻ địch: Solid Small	Đặt một tháp gần đường đi để tấn công kẻ địch.	Kỳ vọng bắn 8 phát đạn thì kẻ địch sẽ chết.	Đạt
2	Cannon Level: 1 Kẻ địch: Car Small	Đặt một tháp gần đường đi để tấn công kẻ địch.	Kỳ vọng bắn một phát thì kẻ địch sẽ chết	Đạt
3	Snipper Level: 1 Kẻ địch: Solid Small	Đặt một tháp gần đường đi để tấn công kẻ địch.	Kỳ vọng bắn xuyên tất cả kẻ địch.	Đạt
4	Time-Wrapper Level: 1 Kẻ địch: Solid Small	Đặt một tháp gần đường đi để tấn công kẻ địch.	Kỳ vọng làm chậm kẻ địch trong phạm vi 20%.	Đạt
5	Laser Level: 1 Kẻ địch: Solid Small	Đặt một tháp gần đường đi để tấn công kẻ địch.	Kỳ vọng tiêu diệt khi laser đốt 1 giây.	Đạt

Bảng 6.4. Kiểm thử sức mạnh tháp

CHƯƠNG 7. VẤN ĐỀ VÀ GIẢI PHÁP

Trong chương trước đã nói về kiểm thử phần mềm, chương này sẽ tiếp tục với những vấn đề gặp phải trong quá trình phát triển.

7.1. Vấn đề gặp phải

Trong quá trình phát triển có một vấn đề ảnh hưởng rất lớn đến hiệu năng gây giật lag. Nguyên nhân do trong game có nhiều đối tượng được sinh ra liên tục với tần suất dày đặc ví dụ như: đạn từ tháp, kẻ địch... điều này ảnh hưởng đến việc dọn dẹp bộ nhớ của Garbage Collector hay gọi tắt là GC.

Đây là hiện tượng bộ nhớ bị phân mảnh và khi càng nhiều đối tượng sinh ra và hủy đi thì tình trạng này càng làm giảm hiệu suất của game.

7.2. Giải pháp

Để giải quyết vấn đề bộ nhớ bị phân mảnh làm giảm hiệu năng của game một giải pháp được đưa ra là sử dụng Object Pooling.

Object Pooling là một kỹ thuật cho phép sử dụng lại những đối tượng đã sản sinh ra trước đó thay vì xóa bỏ đối tượng cũ và tạo ra đối tượng mới. Điều này sẽ giảm thiểu tình trạng bộ nhớ bị phân mảnh.

Cụ thể với dự án. Khi tháp tấn công kẻ địch sẽ tạo ra rất nhiều đạn, thay vì xóa bỏ nó thì em sẽ sử dụng Object Pooling để tái sử dụng đạn bằng cách tắt những viên đạn đã được sử dụng thay vì xóa bỏ chúng và khi nào cần sản sinh ra đạn em sẽ sử dụng lại viên đạn cũ sau đó thay đổi giá trị thuộc tính của viên đạn và tiếp tục sử dụng.

7.3. Kết quả đạt được

Sau khi áp dụng Object Pooling thì hiệu suất của game đã cải thiện đáng kể không còn tình trạng giật lag như lúc chưa sử dụng.

CHƯƠNG 8. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong chương 7 đã trình bày về vấn đề gặp phải trong quá trình hoàn thiện dự án. Ở chương này sẽ là lời kết luận và định hướng phát triển cho dự án trong tương lai

8.1. Kết luận

Trong suốt quá trình thực hiện đồ án game phòng thủ 2D. Nhờ vào những định hướng và chỉ dẫn của ThS. Trần Thanh Hùng, đồ án đã đạt được những thành tựu đáng kể. Em đã xây dựng được một tựa game 2D chiến thuật đồ họa bắt mắt lối chơi hấp dẫn.

đã học được cách để tạo ra một sản phẩm game với đầy đủ các tính năng chính thông qua dự án hiểu được những công đoạn từ lên ý tưởng, thiết kế game kiểm thử...

Em đã học được cách giải quyết vấn đề trong quá trình phát triển trò chơi đã đối mặt với nhiều khó khăn trong việc tối ưu hiệu suất cho game. Đã áp dụng được những kỹ thuật để cải thiện hiệu suất.

Ngoài ra vẫn còn một số vấn đề về mặt giao diện hình ảnh vẫn cần được cải thiện thêm. Kết quả cần được cải thiện thêm trong tương lai.

8.2. Hướng phát triển

Để dự án được hoàn thiện hơn em cần tạo ra nhiều màn chơi hơn nhiều phong cảnh bản đồ đa dạng hơn, tăng thêm số lượng kẻ địch đa dạng hơn về cách chúng tấn công. Tăng thêm khả năng phòng thủ tạo ra nhiều loại tháp hơn cách tấn công đa dạng hơn.

Ngoài ra cần cải thiện hình ảnh bắt mắt và chi tiết hơn tạo ra nhiều hình ảnh đồ họa chất lượng hơn phong phú hơn về thể loại.

Cải thiện hiệu suất cho game là vấn đề rất cần thiết để có thể triển khai được trên những thiết bị có cấu hình thấp. Tối ưu trải nghiệm loại bỏ tình trạng giật lag gây khó chịu cho người chơi.

Trên đây là phần trình bày về kết luận những gì làm được, chưa làm được, những điều đã học hỏi được qua quá trình xây dựng đồ án. Cùng với đó là những hướng phát triển cho đồ án trong tương lai. Đây cũng là chương cuối cùng và là sự tổng kết cho báo cáo và toàn bộ đồ án.

TÀI LIỆU THAM KHẢO

- [1] TheGamer, "Top Game Tower Defense Popular," [Online]. Available: <https://www.thegamer.com/best-tower-defense-games/>. [Accessed 10 05 2024].
- [2] C#, "C Sharp Introduction," [Online]. Available: https://www.w3schools.com/cs/cs_intro.php. [Accessed 12 05 2024].
- [3] Mono, "FrameWork MonoBehavior Unity," [Online]. Available: <https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>. [Accessed 13 05 2024].
- [4] UNITY, "2D Development Game," [Online]. Available: <https://docs.unity3d.com/Manual/Unity2D.html>. [Accessed 14 05 2024].
- [5] C. Campbell, "How to write a video game story," [Online]. Available: <https://www.polygon.com/features/2019/1/10/18165611/how-to-write-a-video-game-story-narrative-building-tips>. [Accessed 11 05 2024].
- [6] OOP, "Object-oriented programming," [Online]. Available: <https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/tutorials/oop>. [Accessed 15 05 2024].
- [7] OOP, "Introduction to object pooling," 15 05 2024. [Online]. Available: <https://learn.unity.com/tutorial/introduction-to-object-pooling>.
- [8] Diagram, "UML Class Diagram Tutorial," [Online]. Available: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/>. [Accessed 11 05 2024].

- [9] Zenva, "Tilemap In Game 2D," [Online]. Available: <https://gamedevacademy.org/mastering-unitys-new-tilemap-editor-building-2d-levels/>. [Accessed 12 05 2024].