

KHOA KỸ THUẬT VÀ CÔNG NGHỆ
BỘ MÔN CÔNG NGHỆ THÔNG TIN



THỰC TẬP ĐỒ ÁN CƠ SỞ NGÀNH
HỌC KỲ I, NĂM HỌC 2024-2025

PHÁT TRIỂN GAME ROGUELIKE DỰA TRÊN NHÂN VẬT GENSIN IMPACT

Giáo viên hướng dẫn
Võ Phước Hưng

Sinh viên thực hiện:
Họ tên: Trần Lâm Phú Đức
Mã số sinh viên: 110122054
Lớp: DA22TTD

Trà Vinh, Tháng 01 Năm 2025

KHOA KỸ THUẬT VÀ CÔNG NGHỆ
BỘ MÔN CÔNG NGHỆ THÔNG TIN



THỰC TẬP ĐỒ ÁN CƠ SỞ NGÀNH
HỌC KỲ I, NĂM HỌC 2024-2025

PHÁT TRIỂN GAME ROGUELIKE DỰA TRÊN NHÂN VẬT GENSIN IMPACT

Giáo viên hướng dẫn
Võ Phước Hưng

Sinh viên thực hiện:
Họ tên: Trần Lâm Phú Đức
Mã số sinh viên: 110122054
Lớp: DA22TTD

Trà Vinh, Tháng 01 Năm 2025

LỜI CẢM ƠN

Trước tiên, tôi xin cảm ơn sâu sắc đến trường Đại học Trà Vinh đã tạo điều kiện để tôi thực hiện dự án này, những người đã đưa ra những sự hỗ trợ, góp ý, cũng như là sự giúp đỡ thầy Võ Phước Hưng – người đóng vai trò quan trọng trong việc phát triển và làm cho dự án thành công tốt đẹp.

Tôi biết ơn tất cả những người đã dành thời công sức, kiến thức và thời gian quý báu của mình để hỗ trợ tôi. Điều đó sẽ là nguồn động lực giúp tôi phát triển bản thân, những ý kiến, lời khuyên đó sẽ góp phần tạo nên một môi trường làm việc tích cực.

Xin chân thành cảm ơn thầy Võ Phước Hưng và trường Đại học Trà Vinh đã giúp đỡ, do đây là dự án đầu tiên cũng như hạn chế về mặt kiến thức lẫn kinh nghiệm nên chắc chắn sẽ xảy ra những sai sót nhỏ, vậy nên tôi mong rằng sẽ nhận được những góp ý và lời khuyên, tôi sẽ đưa những lời khuyên, góp ý đó vào dự án này để tạo nên một thành công tốt đẹp.

Trân trọng,

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN NGHIÊN CỨU.....	1
1.1. Lý do chọn đề tài	1
1.2. Giới thiệu đề tài.....	1
1.3. Mục đích nghiên cứu.....	2
1.4. Đối tượng nghiên cứu.....	2
1.4.1 Lối chơi của Roguelike	3
1.4.2 Nhân vật từ Genshin Impact.....	3
1.4.3 Công cụ phát triển GameMaker	3
1.4.4 Yếu tố tương tác và trải nghiệm người chơi	3
1.4.5 Hệ thống vật phẩm và nâng cấp	3
1.5. Phạm vi nghiên cứu	3
1.5.1 Công nghệ sử dụng	3
1.5.2 Chức năng ứng dụng.....	4
1.5.3 Quy mô nghiên cứu.....	5
1.5.4 Khả năng nghiên cứu và áp dụng.....	5
1.6. Phương pháp nghiên cứu	5
CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT	7
2.1. Lý thuyết về thể loại Roguelike	7
2.1.1 Permadeath (Chết vĩnh viễn).....	7
2.1.2 Tiến trình dựa trên kỹ năng và quyết định của người chơi.....	7
2.2. Các yếu tố thiết kế game trong thể loại Roguelike.....	7
2.2.1 Hệ thống chiến đấu trong game.....	7
2.2.2 Hệ thống vật phẩm và nâng cấp.....	8
2.2.3 Kẻ thù và AI trong game	8

2.3. Lý thuyết về phát triển game với GameMaker	8
2.3.1 Project.....	8
2.3.2 Room.....	9
2.3.3 Objects.....	9
2.3.4 Sprites	10
2.3.5 Event.....	10
2.3.6 Sounds	11
2.3.7 Tilesets.....	11
CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU	12
3.1. Chuẩn bị.....	12
3.2. Tạo Project, nhân vật và lập trình di chuyển cho nhân vật.....	12
3.2.1 Tạo Project.....	12
3.2.2 Tạo nhân vật	13
3.3. Tạo map chơi, giao diện và cơ chế cho Menu, Chọn nhân vật.....	17
3.3.1 Tạo map chơi	17
3.3.2 Tạo giao diện và cơ chế cho Menu	21
3.3.3 Tạo giao diện và cơ chế chọn nhân vật	23
3.4. Tạo quái, boss và cơ chế tấn công của nhân vật.....	26
3.4.1 Tạo quái và boss.....	26
3.4.2 Cơ chế tấn công.....	27
3.5. Quái xuất hiện ngẫu nhiên và bộ đếm thời gian	28
3.5.1 Quái xuất hiện.....	28
3.5.2 Bộ đếm thời gian	29
3.6. Cơ chế lên cấp và hệ thống nâng cấp kỹ năng.....	31
3.6.1 Cơ chế lên cấp	31

3.6.2 Hệ thống nâng cấp kỹ năng.....	31
3.7. Thêm nhạc nền, âm thanh hiệu ứng và giao diện tùy chỉnh.....	35
3.7.1 Thêm nhạc nền và âm thanh hiệu ứng.....	35
3.7.2 Giao diện và hệ thống tùy chỉnh	36
3.8. Hệ thống nâng cấp bằng tiền.....	36
CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU	38
4.1. Tổng quan kết quả	38
4.2. Kết quả đạt được	38
4.2.1 Giao diện Menu chính.....	38
4.2.2 Cơ chế và giao diện chọn nhân vật.....	38
4.2.3 Cơ chế và giao diện nâng cấp.....	39
4.2.4 Cơ chế và giao diện tùy chỉnh.....	39
4.2.5 Cơ chế chiến đấu	40
4.2.6 Cơ chế lên cấp và chọn kỹ năng.....	41
4.2.7 Kẻ thù xuất hiện theo thời gian	42
4.2.8 Cơ chế sinh tồn.....	43
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	44
5.1. Kết luận.....	44
5.2. Hướng phát triển.....	44

DANH MỤC HÌNH ẢNH

Hình 1.1 GameMaker.....	4
Hình 1.2 Aseprite.....	4
Hình 2.1 Giao diện tạo Project.....	8
Hình 2.2 Giao diện Room.....	9
Hình 2.3 Giao diện Object.....	9
Hình 2.4 Giao diện Sprite.....	10
Hình 3.1 Lưu đồ thuật toán	12
Hình 3.2 Tạo Project 1	13
Hình 3.3	13
Hình 3.4 Tạo Object.....	14
Hình 3.5 Thêm Event	14
Hình 3.6 Lệnh di chuyển	15
Hình 3.7 Tạo Sprite	15
Hình 3.8 Thêm hình ảnh vào Sprite	16
Hình 3.9 Gán Sprite cho Object.....	16
Hình 3.10 Camera theo nhân vật	17
Hình 3.11 Camera theo nhân vật	17
Hình 3.12 Camera theo nhân vật	17
Hình 3.13 Tileset	18
Hình 3.14 Tạo Tileset.....	18
Hình 3.15 Thêm Sprite vào Tileset	19
Hình 3.16 Thêm Tileset vào Room	19
Hình 3.17 Thêm Tileset vào Room	20
Hình 3.18 Map Chơi.....	20

Hình 3.19 Tạo Room.....	21
Hình 3.20 Thêm Backgroud cho Room.....	21
Hình 3.21 Thêm nút điều hướng vào Room.....	22
Hình 3.22 Thêm các nút điều hướng vào Room.....	22
Hình 3.23 Lệnh chuyển Room.....	22
Hình 3.24 Lệnh phóng to nút	23
Hình 3.25 Lện thu nhỏ nút.....	23
Hình 3.26 Đổi Room đầu tiên.....	23
Hình 3.27 Menu chọn nhân vật.....	24
Hình 3.28 Thẻ chọn nhân vật.....	24
Hình 3.29 Lệnh chọn nhân vật.....	25
Hình 3.30 Thêm các nhân vật vào Parent.....	25
Hình 3.31 Lệnh tạo nhân vật được chọn.....	26
Hình 3.32 Room chọn nhân vật.....	26
Hình 3.33 Lệnh quái theo nhân vật	26
Hình 3.34 Lệnh quái gây sát thương	27
Hình 3.35. Lệnh gây sát thương cho quái.....	27
Hình 3.36 Các biến Alarm.....	28
Hình 3.37 Lệnh kỹ năng	28
Hình 3.38 Biến lưu vị trí xuất hiện của quái.....	29
Hình 3.39 Lệnh cho quái xuất hiện ngẫu nhiên	29
Hình 3.40 Các biến của bộ đếm thời gian	30
Hình 3.41 Lệnh của bộ đếm thời gian.....	30
Hình 3.42 Lệnh vẽ bộ đếm thời gian.....	30
Hình 3.43 Lệnh nhật kinh nghiệm.....	31

Hình 3.44	Lệnh lên cấp.....	31
Hình 3.45	Lệnh vẽ thanh kinh nghiệm.....	31
Hình 3.46	Sprite của các kỹ năng.....	32
Hình 3.47	Biến lưu các Sprite của kỹ năng.....	32
Hình 3.48	Biến cho hàm chọn ngẫu nhiên	33
Hình 3.49	Lệnh chọn ra 3 số ngẫu nhiên.....	33
Hình 3.50	Lệnh in ra 3 Sprite.....	34
Hình 3.51	Lệnh nâng cấp các kỹ năng.....	34
Hình 3.52	Tạo Sound	35
Hình 3.53	Lệnh phát nhạc nền.....	35
Hình 3.54	Lệnh tang âm lượng nhạc nền	36
Hình 3.55	Lệnh hiển thị nhạc nền	36
Hình 3.56	Lệnh nút nâng cấp.....	37
Hình 3.57	Lệnh in dòng txt	37
Hình 3.58	Lệnh hiển thị thông số và tiền hiện có.....	37
Hình 4.1	Giao diện Menu chính	38
Hình 4.2	Giao diện chọn nhân vật.....	39
Hình 4.3	Giao diện nâng cấp.....	39
Hình 4.4	Giao diện tùy chỉnh.....	40
Hình 4.5	Kỹ năng riêng của nhân vật 1	40
Hình 4.6	Kỹ năng riêng của nhân vật 2	41
Hình 4.7	Kỹ năng riêng của nhân vật 3	41
Hình 4.8	Cơ chế rút kinh nghiệm khi tiêu diệt quái.....	42
Hình 4.9	Thanh kinh nghiệm	42
Hình 4.10	Cơ chế lên cấp và chọn kỹ năng.....	42

Hình 4.11 Cơ chế quái xuất hiện ngẫu nhiên	43
Hình 4.12 Khi nhân vật chết	43
Hình 4.13 Khi tiêu diệt được boss cuối	43

CHƯƠNG 1. TỔNG QUAN NGHIÊN CỨU

1.1. Lý do chọn đề tài

Roguelike là một thể loại đang thu hút sự quan tâm lớn từ cộng đồng game thủ nhờ vào yếu tố ngẫu nhiên, độ thử thách cao, và tính lặp lại hấp dẫn trong từng lần chơi. Đặc biệt, từ năm 2022 đến nay với sự thành công của các tựa game như *Holocure*, lấy cảm hứng từ các nhân vật VTuber, hay chế độ *Thảm Họa Đỏ Bộ* (Invasion Mode) trong *Liên Minh Huyền Thoại*, đã chứng minh tiềm năng kết hợp các nhân vật quen thuộc với cơ chế **Roguelike** có thể tạo ra trải nghiệm game độc đáo và mới lạ thu hút người chơi.

Dựa trên các tiền đề đó, lựa chọn đề tài **Phát triển game Roguelike dựa trên nhân vật Genshin Impact** vừa tận dụng được độ nổi tiếng của các nhân vật trong *Genshin Impact* kết hợp với sức hút của thể loại Roguelike để mang lại trải nghiệm mới mẻ và hấp dẫn cho người chơi. Đồng thời đây cũng là cơ hội được thực hành, nắm bắt, trải nghiệm quá trình nghiên cứu, thiết kế, lập trình và tối ưu hóa để tạo ra được một con game.

1.2. Giới thiệu đề tài

Đề tài "Phát triển game **Roguelike** lấy nhân vật từ *Genshin Impact* bằng **GameMaker**" không chỉ kế thừa các yếu tố đặc trưng của thể loại **Roguelike** mà còn tận dụng sức hút từ các nhân vật nổi tiếng trong *Genshin Impact*, một trò chơi nổi bật với hệ thống nhân vật đa dạng và cốt truyện sâu sắc.

Điểm nổi bật của đề tài là việc tái hiện các nhân vật từ *Genshin Impact* dưới dạng pixel art, phù hợp với phong cách 2D, đồng thời giữ nguyên các đặc trưng về kỹ năng và phong cách chiến đấu của từng nhân vật. Trò chơi sẽ mang đến cho người chơi một trải nghiệm mới lạ, kết hợp giữa tính ngẫu nhiên của **Roguelike** và các nhân vật từ *Genshin Impact*.

Sử dụng **GameMaker** làm công cụ phát triển chính, đề tài tận dụng sức mạnh của nền tảng này trong việc xây dựng các cơ chế game như di chuyển, chiến đấu, và nâng cấp, đồng thời tối ưu hóa hiệu suất cho các màn chơi có số lượng quái vật lớn.

Bằng việc thực hiện đề tài này, nhóm nghiên cứu không chỉ nâng cao hiểu biết về quy trình phát triển game mà còn rèn luyện kỹ năng lập trình, thiết kế đồ họa pixel, và quản lý dự án. Kết quả của đề tài là một sản phẩm mẫu, có tiềm năng mở rộng và phát triển thêm trong tương lai.

1.3. Mục đích nghiên cứu

Mục đích của đề tài là xây dựng một trò chơi **Roguelike** kết hợp yếu tố nhân vật từ *Genshin Impact* tạo ra một game giải trí mới lạ, phù hợp với xu hướng và sở thích của người chơi hiện nay. Trò chơi không chỉ mang lại trải nghiệm thú vị mà còn là bước đầu tiên cho việc thực hiện hóa ước mơ trở thành một lập trình game của tôi.

Cụ thể:

Xây dựng hệ thống nhân vật độc đáo: Tích hợp các nhân vật từ *Genshin Impact* dưới dạng đồ họa pixel art, đồng thời giữ nguyên hoặc điều chỉnh kỹ năng đặc trưng để phù hợp với phong cách **Roguelike**.

Phát triển cơ chế gameplay hấp dẫn: Xây dựng các tính năng chính như lựa chọn nhân vật, hệ thống chiến đấu với quái vật, hệ thống vật phẩm ngẫu nhiên, và cơ chế nâng cấp phù hợp với thể loại **Roguelike**.

Trải nghiệm người chơi: Thiết kế giao diện trực quan, thân thiện với người dùng, đảm bảo tính mượt mà trong quá trình điều khiển nhân vật và tương tác với các yếu tố trong trò chơi.

Khả năng ứng dụng công nghệ: Sử dụng **GameMaker** để phát triển game, tận dụng các tính năng như lập trình logic, xử lý đồ họa 2D, và quản lý tài nguyên hiệu quả. Đảm bảo việc tối ưu hóa hiệu năng game và đảm bảo tính linh hoạt để mở rộng trong tương lai.

1.4. Đối tượng nghiên cứu

Nghiên cứu tập trung vào các yếu tố chính liên quan đến việc xây dựng và phát triển trò chơi **Roguelike** kết hợp với nhân vật từ *Genshin Impact*. Cụ thể, các đối tượng nghiên cứu bao gồm:

1.4.1 Lối chơi của Roguelike

Tính ngẫu nhiên: Quái vật và vật phẩm được tạo ra ngẫu nhiên, mang lại trải nghiệm mới mẻ mỗi lần chơi.

Độ khó cao: Người chơi thường chỉ có một cơ hội sinh tồn, và khi thất bại sẽ phải bắt đầu lại từ đầu (permadeath).

Tiến trình dựa trên kỹ năng: Thành công của người chơi phụ thuộc vào sự thành thạo cơ chế game và khả năng thích nghi với các tình huống mới.

Lối chơi này không chỉ tạo ra sự thử thách mà còn mang đến cảm giác hào hứng và thỏa mãn khi người chơi vượt qua được các trở ngại.

1.4.2 Nhân vật từ Genshin Impact

Tập trung vào các nhân vật nổi bật từ *Genshin Impact*, bao gồm đặc điểm, kỹ năng và phong cách chiến đấu. Các nhân vật sẽ được thiết kế lại dưới dạng pixel art nhưng vẫn giữ nguyên nét đặc trưng của họ để tạo sự quen thuộc và hấp dẫn cho người chơi.

1.4.3 Công cụ phát triển GameMaker

GameMaker được lựa chọn làm công cụ chính để lập trình và thiết kế trò chơi, nhờ tính linh hoạt và hỗ trợ mạnh mẽ trong việc phát triển game 2D.

1.4.4 Yếu tố tương tác và trải nghiệm người chơi

Nghiên cứu cách xây dựng giao diện, điều khiển và các yếu tố tương tác để mang lại trải nghiệm tốt nhất cho người chơi.

1.4.5 Hệ thống vật phẩm và nâng cấp

Xây dựng hệ thống vật phẩm, kỹ năng nâng cấp nhân vật, và các yếu tố ngẫu nhiên ảnh hưởng đến chiến thuật chơi.

1.5. Phạm vi nghiên cứu

1.5.1 Công nghệ sử dụng

GameMaker: Là một engine cho phép các nhà phát triển tạo ra các game 2D và 3D.



Hình 1.1 GameMaker

Aseprite: Là phần mềm dùng để vẽ nhân vật môi trường dưới dạng pixel, 2D art.



Hình 1.2 Aseprite

1.5.2 Chức năng ứng dụng

Chọn nhân vật: Người chơi sẽ chọn một nhân vật *Genshin Impact* trong số nhân vật có sẵn để sử dụng. Mỗi nhân vật có bộ kỹ năng riêng biệt.

Cơ chế chiến đấu: Nhân vật sẽ tự động tấn công kẻ thù theo hướng chỉ định của người chơi, người chơi có thể điều khiển nhân vật né tránh và sử dụng kỹ năng đặc biệt để hỗ trợ tiêu diệt kẻ địch.

Hệ thống nâng cấp kỹ năng: Khi tiêu diệt kẻ thù, nhân vật sẽ thu thập điểm kinh nghiệm để lên cấp. Khi lên cấp, người chơi nhận được các thẻ vật phẩm để nhân thêm kỹ năng hoặc nâng cấp các kỹ năng hiện có. Các cấp độ cao hơn sẽ yêu cầu nhiều điểm kinh nghiệm hơn.

Kẻ thù xuất hiện theo thời gian: Kẻ thù sẽ xuất hiện theo thời gian sẽ tăng dần về số lượng và độ khó. Sau một khoảng thời gian nhất định, sẽ có boss xuất hiện. Người chơi tiêu diệt boss sẽ nhận được nhiều kinh nghiệm hơn.

Cơ chế sinh tồn: Người chơi sẽ chơi liên tục cho đến khi nhân vật bị đánh bại hoặc tiêu diệt được boss cuối.

1.5.3 Quy mô nghiên cứu

Dự án tập trung phát triển một game mẫu với một số lượng giới hạn nhân vật, quái vật, và vật phẩm để đảm bảo hoàn thiện trong thời gian ngắn.

1.5.4 Khả năng nghiên cứu và áp dụng

Kết quả nghiên cứu có thể được sử dụng làm nền tảng để phát triển các tựa game tương tự hoặc mở rộng tính năng cho game sau khi hoàn thành.

1.6. Phương pháp nghiên cứu

Để phát triển trò chơi **Roguelike** sử dụng nhân vật từ *Genshin Impact*, nghiên cứu này sẽ thực hiện qua ba giai đoạn chính: lập kế hoạch, phát triển trên **GameMaker** và kiểm thử.

Giai đoạn 1: Lập kế hoạch và thiết kế

Lên ý tưởng: Xác định rõ cơ chế game, bao gồm các yếu tố ngẫu nhiên, hệ thống chiến đấu, và cơ chế nâng cấp. Nhân vật từ *Genshin Impact* sẽ được lựa chọn và tái hiện trong game, kèm theo việc xây dựng hệ thống kỹ năng đặc trưng cho từng nhân vật.

Thiết kế đồ họa và âm thanh: Thiết kế đồ họa cho các nhân vật, kẻ thù, và môi trường game dưới dạng pixel art, đồng thời xây dựng các hiệu ứng âm thanh phù hợp để tạo ra không khí cho trò chơi. Các yếu tố âm thanh sẽ bao gồm âm thanh chiến đấu, âm thanh bước chân, và nhạc nền.

Thiết kế các tính năng chính: Xác định và vẽ lưu đồ cho các tính năng quan trọng như hệ thống chiến đấu, cơ chế nâng cấp nhân vật, và cơ chế sinh tồn (permadeath). Các tính năng này sẽ quyết định trải nghiệm người chơi và tạo nên sự hấp dẫn cho trò chơi.

Giai đoạn 2: Phát triển trên GameMaker

Lập trình cơ chế di chuyển và chiến đấu: Lập trình cơ chế di chuyển cho nhân vật, cho phép nhân vật di chuyển tự do trong môi trường game và tự động tấn công khi gặp kẻ thù. Cơ chế chiến đấu sẽ được thiết kế để đơn giản nhưng đầy đủ các tính năng chiến đấu của nhân vật từ *Genshin Impact*.

Tạo hệ thống nâng cấp: Xây dựng hệ thống để khi nhân vật đạt đủ điểm kinh nghiệm, người chơi có thể lên cấp và nhận vật phẩm có thể nâng cấp các kỹ năng hoặc đặc điểm của nhân vật.

Tạo kẻ thù và boss: Phát triển AI cho kẻ thù với hành vi đa dạng như di chuyển, tấn công, và tránh né. Các boss cũng sẽ được tạo ra với cơ chế tấn công riêng biệt và các chiêu thức đặc trưng, tạo thử thách cho người chơi.

Tích hợp âm thanh và hiệu ứng: Chèn các hiệu ứng âm thanh và hình ảnh vào trò chơi để tăng tính hấp dẫn và tạo sự sống động cho các hành động trong game như chiến đấu, thăng cấp, và các sự kiện đặc biệt.

Giai đoạn 3: Kiểm thử và sửa lỗi

Kiểm thử gameplay: Tiến hành chạy thử game để kiểm tra tính cân bằng của các yếu tố trong game, bao gồm hệ thống kẻ thù, boss, và nhân vật. Điều chỉnh các yếu tố như sát thương, tốc độ xuất hiện của kẻ thù, khả năng sinh tồn của nhân vật, và hiệu quả của hệ thống nâng cấp để đảm bảo trải nghiệm chơi thú vị và công bằng.

Xác định lỗi và sửa lỗi: Chạy game nhiều lần để kiểm tra các tương tác giữa các đối tượng trong game, tìm kiếm các bug và lỗi. Sau khi xác định lỗi, tiến hành sửa chữa và tinh chỉnh các lỗi đã phát hiện để nâng cao chất lượng trò chơi.

CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT

2.1. Lý thuyết về thể loại Roguelike

Roguelike là một thể loại trò chơi video đặc trưng bởi việc kết hợp các yếu tố ngẫu nhiên trong thiết kế trò chơi, cùng với việc nhấn mạnh tính thử thách và khả năng tái chơi cao. Các yếu tố quan trọng của thể loại này bao gồm:

2.1.1 Permadeath (Chết vĩnh viễn)

Một yếu tố không thể thiếu trong Roguelike là "permadeath", tức là khi nhân vật chết, người chơi sẽ không thể tiếp tục từ điểm đã chết mà phải bắt đầu lại từ đầu. Điều này tạo ra một áp lực cao đối với người chơi, yêu cầu họ phải quản lý tài nguyên và chiến đấu một cách khôn ngoan.

2.1.2 Tiến trình dựa trên kỹ năng và quyết định của người chơi

Trong trò chơi roguelike, sự thành công của người chơi không chỉ phụ thuộc vào may mắn mà còn vào kỹ năng, chiến lược và các quyết định mà họ đưa ra trong suốt quá trình chơi. Các yếu tố này được thể hiện rõ qua việc chọn lựa vật phẩm, nâng cấp kỹ năng và đối phó với các tình huống khó khăn.

2.2. Các yếu tố thiết kế game trong thể loại Roguelike

Để tạo ra một trò chơi roguelike hấp dẫn và thú vị, việc áp dụng các yếu tố thiết kế game là vô cùng quan trọng. Các yếu tố này không chỉ ảnh hưởng đến cách thức chơi mà còn quyết định đến trải nghiệm của người chơi.

2.2.1 Hệ thống chiến đấu trong game

Trong trò chơi **Roguelike**, hệ thống chiến đấu thường đơn giản nhưng đầy thử thách. Các nhân vật có thể tấn công và sử dụng kỹ năng để chiến đấu với kẻ thù. Mỗi nhân vật trong trò chơi sẽ có những kỹ năng đặc trưng, làm phong phú thêm tính chiến thuật của trò chơi. Trong đề án này, các nhân vật từ *Genshin Impact* sẽ được tích hợp với cơ chế chiến đấu đặc trưng của **Roguelike**, giúp tạo ra một trải nghiệm độc đáo.

2.2.2 Hệ thống vật phẩm và nâng cấp

Hệ thống vật phẩm và nâng cấp đóng vai trò quan trọng trong các trò chơi **Roguelike**. Vật phẩm có thể là các loại vũ khí, trang bị hoặc các vật dụng hỗ trợ khác, giúp người chơi cải thiện sức mạnh và khả năng sống sót. Việc nâng cấp kỹ năng nhân vật khi đạt đủ điểm kinh nghiệm sẽ giúp người chơi cảm thấy tiến bộ qua từng màn chơi và thêm động lực để tiếp tục.

2.2.3 Kẻ thù và AI trong game

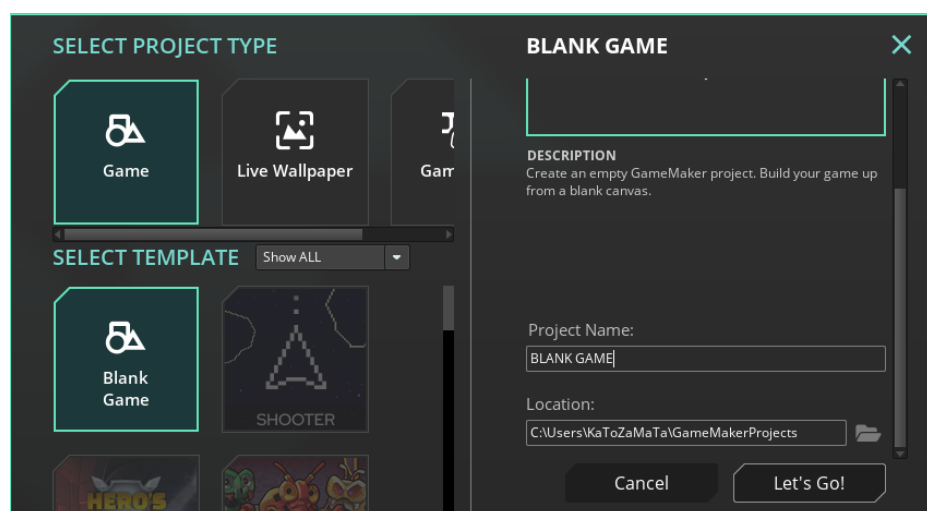
Kẻ thù trong trò chơi **Roguelike** thường đa dạng về hành vi và chiến thuật. AI của kẻ thù phải đủ thông minh để tạo ra thử thách cho người chơi. Trong trò chơi này, các loại kẻ thù sẽ được thiết kế với hành vi chiến đấu đặc trưng và có thể thay đổi theo thời gian, tạo nên sự bất ngờ và thách thức cho người chơi.

2.3. Lý thuyết về phát triển game với GameMaker

GameMaker là một engine cho phép các nhà phát triển tạo ra các game 2D và 3D. GameMaker sử dụng ngôn ngữ lập trình GML (Game Maker Language), được nhiều người sử dụng.

2.3.1 Project

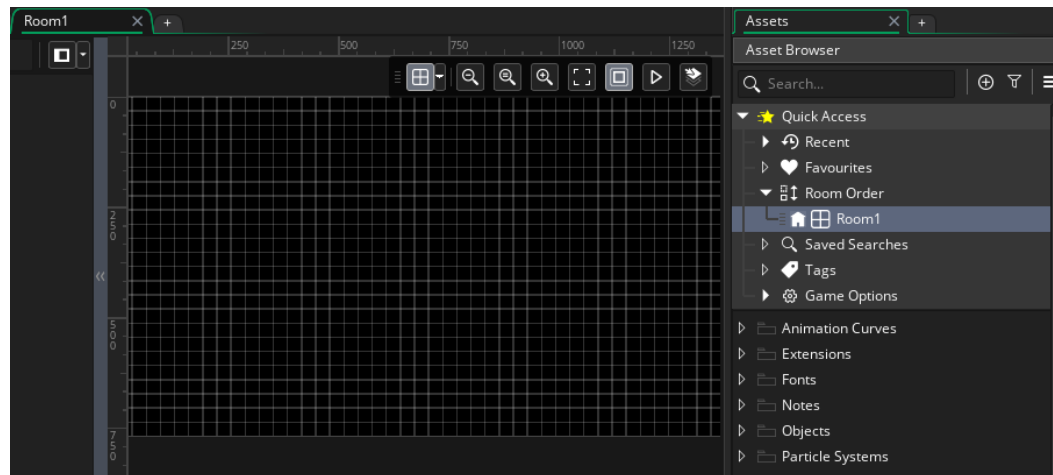
Project (dự án) là nơi chứa toàn bộ các thành phần tạo nên một tựa game hoàn chỉnh: room (phòng), objects (đối tượng), Sprites (hình ảnh, hoạt ảnh),...



Hình 2.1 Giao diện tạo Project

2.3.2 Room

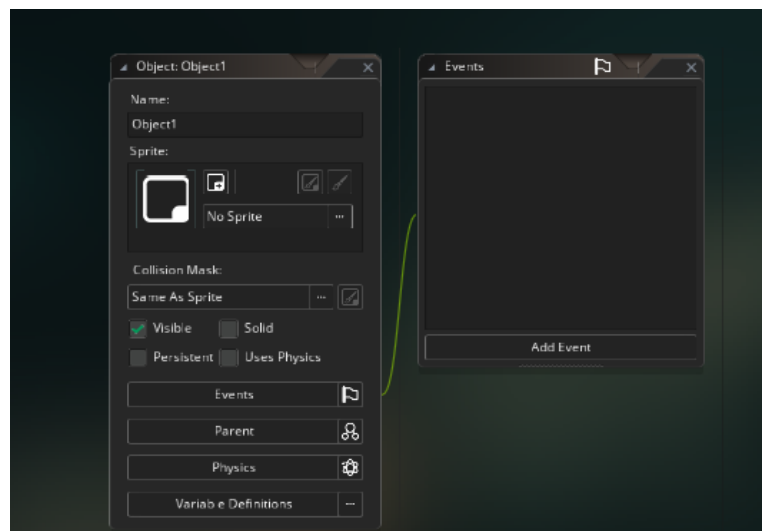
Được hiểu là một nền chứa tập hợp các nút, các vật thể, các dòng chữ, ... xuất hiện trong game dưới dạng giao diện đồ họa. Một Project phải có nhiều hơn hoặc bằng 1 Room, trong đó "Room1" hiển thị đầu tiên khi game chạy, nó có thể là các level, các menu, các cửa sổ trò chơi khác nhau, ...



Hình 2.2 Giao diện Room

2.3.3 Objects

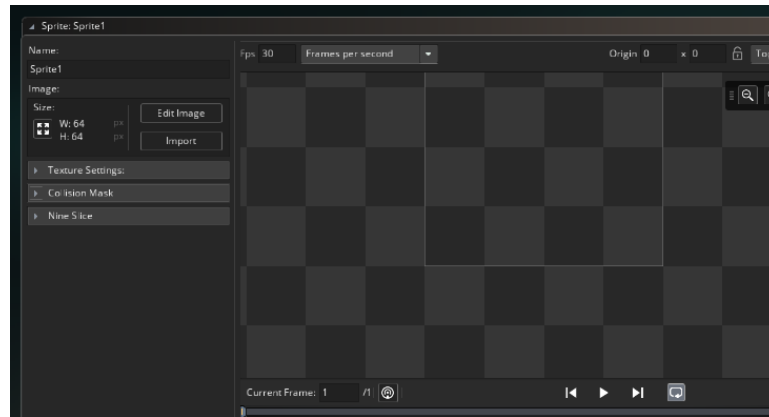
Là các thành phần chính trong game, điểm khác biệt lớn nhất của nó và các thành phần khác là người chơi có thể tương tác với các Object, và diễn biến trong trò chơi sẽ thay đổi dựa vào sự tương tác này, nó có thể là 1 cú click chuột, 1 thao tác ấn phím,... dựa vào các Event (sự kiện)



Hình 2.3 Giao diện Object

2.3.4 Sprites

Là tài nguyên đồ họa, nó có thể là các ảnh nhân vật, ảnh nền, đồ họa cho các nút hay các object trong game, hoặc thậm chí là ảnh động. Các ảnh này có thể là tự vẽ hoặc tải về từ trên mạng.



Hình 2.4 Giao diện Sprite

2.3.5 Event

Là các tình huống hoặc hành động xảy ra trong trò chơi do người dùng lập trình để đối tượng (object) phản hồi. Event đóng vai trò như một "kích hoạt" để điều khiển hành vi của các đối tượng, chẳng hạn như khi một đối tượng va chạm với đối tượng khác, khi người chơi nhấn phím, hoặc khi một khoảng thời gian trôi qua. Các Event trong Gamemaker:

Create Event (Sự kiện khởi tạo): Kích hoạt khi đối tượng được tạo (xuất hiện) trong trò chơi. Thường dùng để khởi tạo giá trị ban đầu cho các biến hoặc thiết lập trạng thái ban đầu của đối tượng.

Step Event (Sự kiện bước): Xảy ra trong mỗi khung hình (frame) của trò chơi. Chia làm ba loại:

Begin Step: Thực hiện trước khi các đối tượng khác cập nhật trạng thái.

Step: Diễn ra trong quá trình xử lý logic chính.

End Step: Thực hiện sau khi các đối tượng khác đã cập nhật trạng thái.

Collision Event (Sự kiện va chạm): Xảy ra khi một đối tượng va chạm với một đối tượng khác. Dùng để xử lý các tương tác như làm mất máu khi kẻ địch chạm vào người chơi, hoặc thu thập vật phẩm khi nhân vật va chạm với chúng.

Alarm Event (Sự kiện báo thức): Kích hoạt sau một khoảng thời gian được đặt trước bằng cách sử dụng các biến alarm (alarm[0], alarm[1], ...). Thường dùng để thực hiện hành động sau một khoảng thời gian, như hồi máu, bắn đạn, hoặc tạo kẻ thù mới.

Keyboard Event (Sự kiện bàn phím): Xảy ra khi người chơi nhấn hoặc thả một phím cụ thể.

Mouse Event (Sự kiện chuột): Xảy ra khi người chơi tương tác bằng chuột, chẳng hạn như nhấn hoặc di chuyển chuột qua đối tượng.

Global Mouse Events: Nhấn chuột ở bất kỳ vị trí nào trong room, không cần nằm trên đối tượng.

Draw Event (Sự kiện vẽ): Xảy ra khi đối tượng được vẽ lên màn hình, Thường dùng để tùy chỉnh cách đối tượng hiển thị, như thay đổi màu sắc, xoay, hoặc hiển thị văn bản kèm theo.

Other Events (Các sự kiện khác): Bao gồm nhiều sự kiện đặc biệt như:

Outside Room: Khi đối tượng đi ra khỏi ranh giới của room.

Game Start/Game End: Xảy ra khi trò chơi bắt đầu hoặc kết thúc.

Room Start/Room End: Khi một room bắt đầu hoặc kết thúc.

2.3.6 Sounds

Các hiệu ứng âm thanh (SFX) hoặc nhạc nền (BGM) trong trò chơi.

2.3.7 Tilesets

Là tập hợp các hình ảnh nhỏ (tile) được dùng để tạo bản đồ hoặc môi trường trong game, đặc biệt là trong các game 2D.

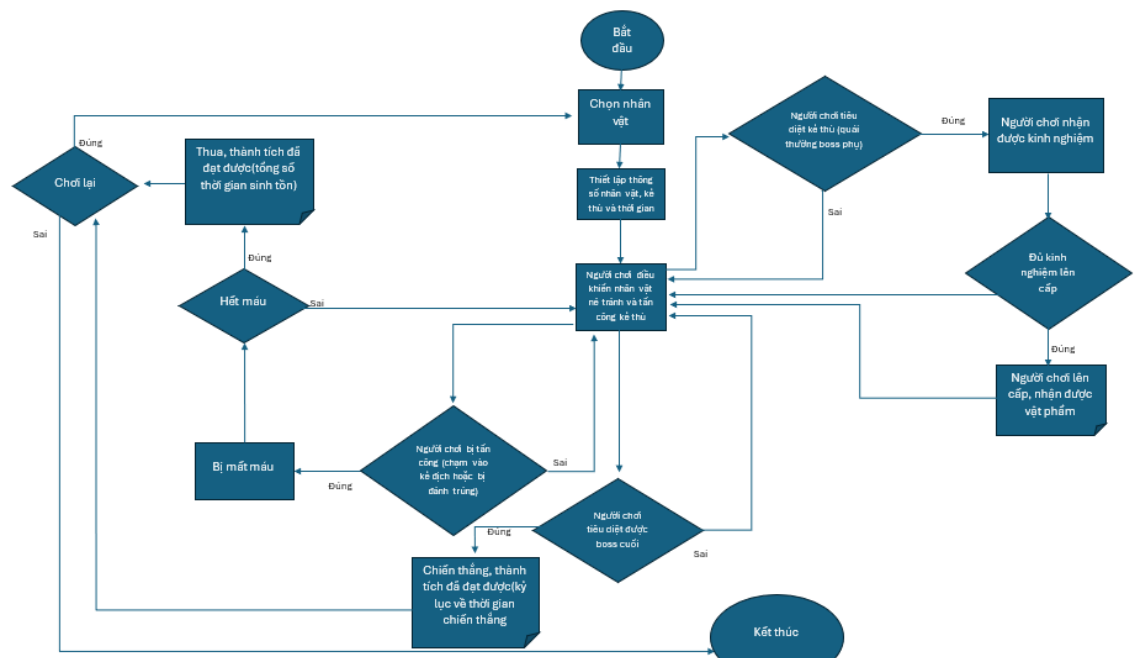
CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU

3.1. Chuẩn bị

Nắm rõ các chế Gameplay của dòng game Roguelike.

Chọn nhân vật phù hợp với kỹ năng đơn giản để đưa vào game.

Vẽ sơ lưu đồ thuật toán toàn diện cho game



Hình 3.1 Lưu đồ thuật toán

3.2. Tạo Project, nhân vật và lập trình di chuyển cho nhân vật

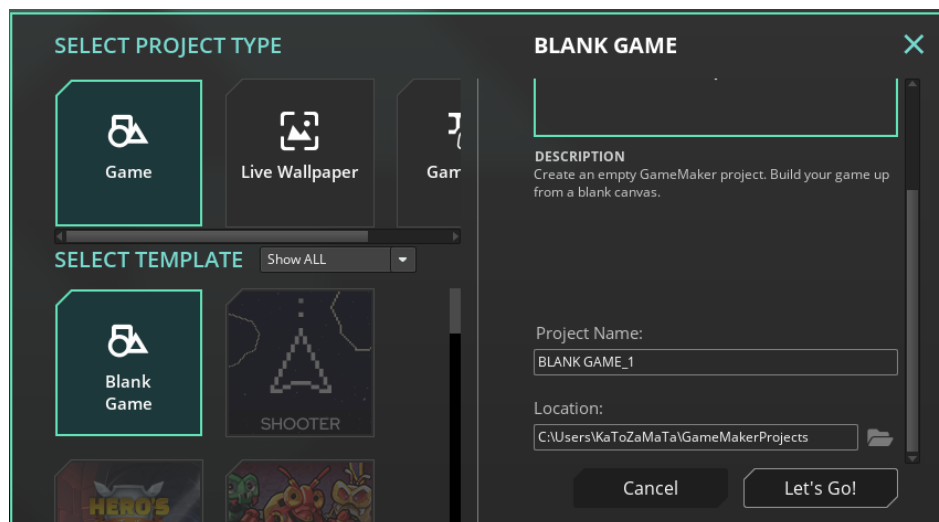
3.2.1 Tạo Project

Tại giao diện chính của GameMaker bấm New.



Hình 3.2 Tạo Project 1

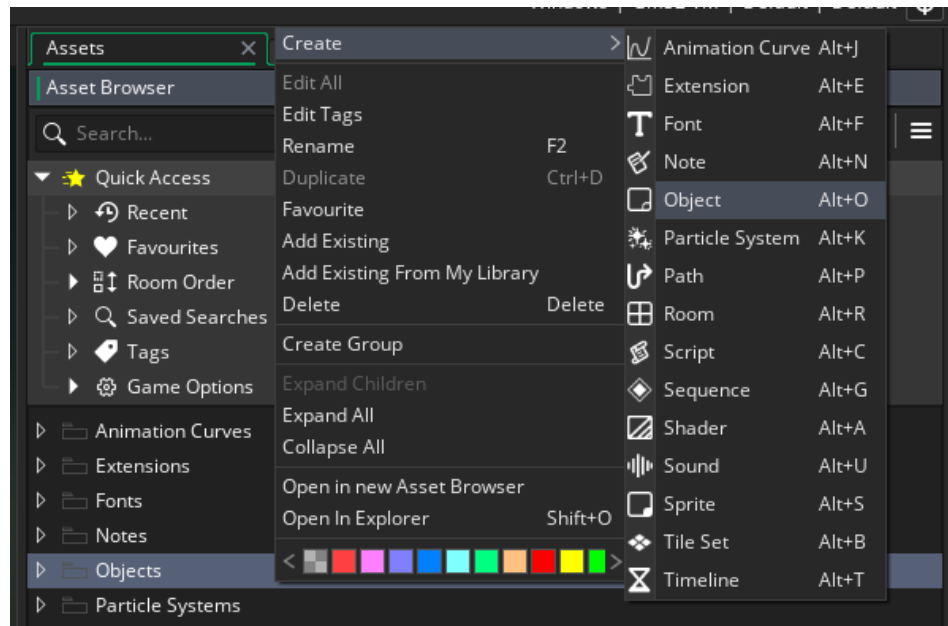
Sau đó, tại mục SELECT PROJECT TYPE chọn Game, mục SELECT TEMPLATE chọn Blank Game. Ở mục Project Name đặt tên game, mục Location chọn vị trí lưu src game.



Hình 3.3

3.2.2 Tạo nhân vật

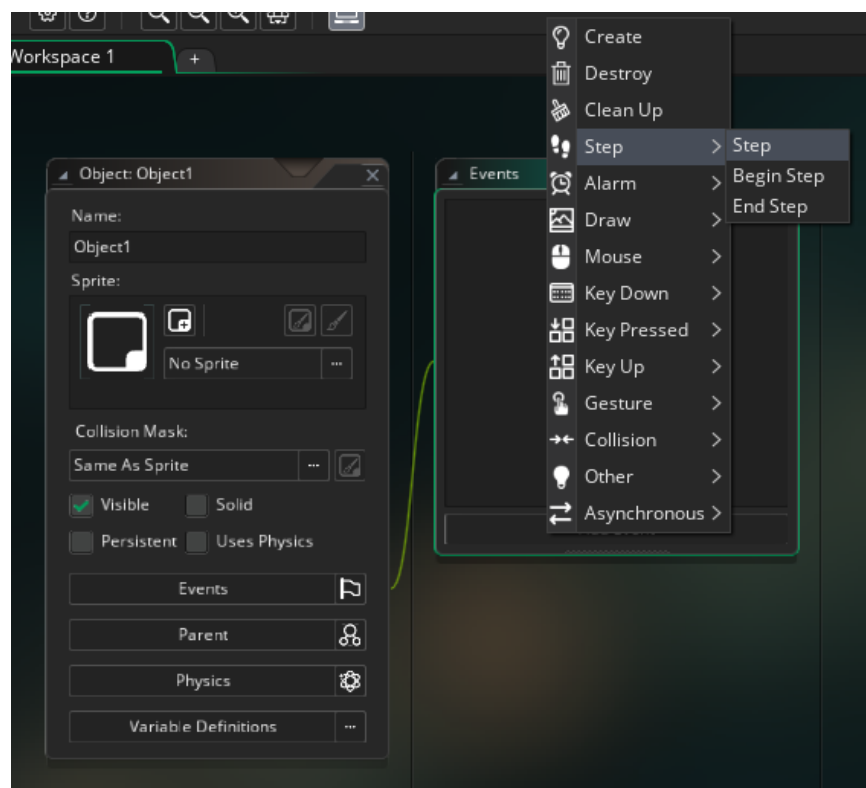
Tại giao diện của Project nhấp chuột phải vào Object, chọn Create, rồi chọn Object để tạo Object cho nhân vật.



Hình 3.4 Tạo Object

Sau khi tạo xong Object, thêm các sự kiện thích hợp để điều khiển nhân vật di chuyển, ở đây dùng Event Step.

Tại Object vừa tạo chọn Add Event, chọn Step, Step để tạo Event Step



Hình 3.5 Thêm Event

Sau khi tạo Event Step tiến lập trình cho nhân vật di chuyển

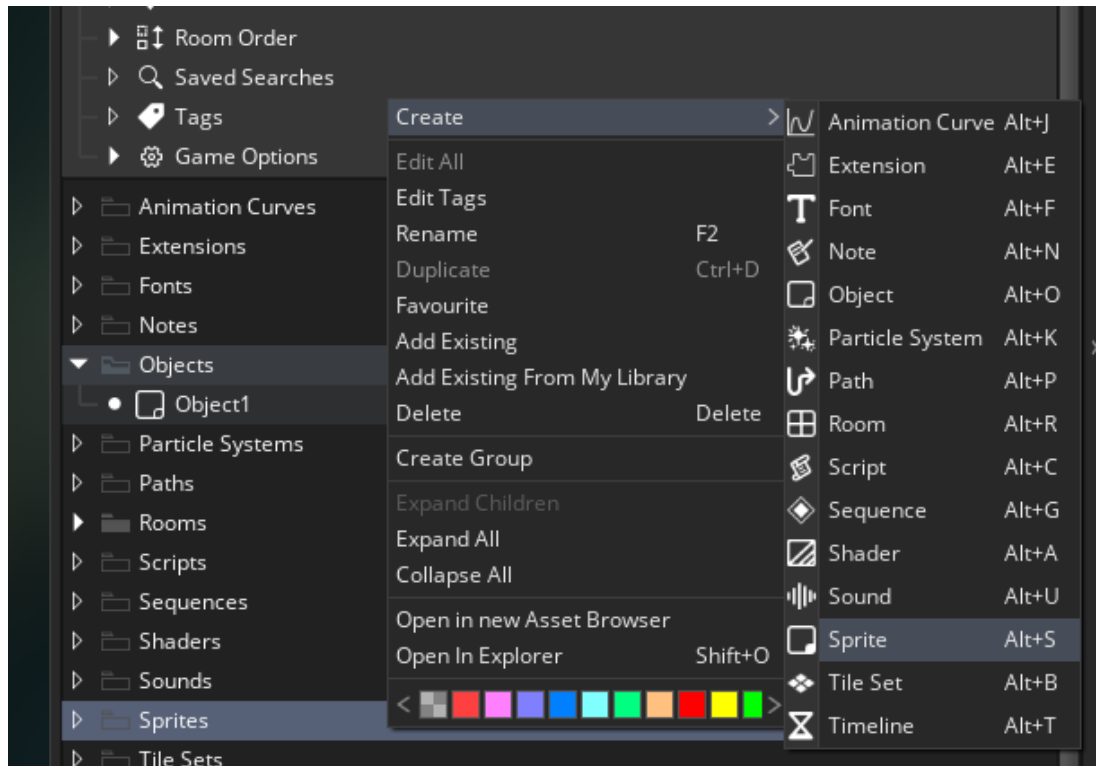
Sử dụng keyboard(phím gán vào) để trả về True nếu người chơi bấm vào phím được gán vào.

```
var left = keyboard_check(vk_left) or keyboard_check(ord("A"));  
var right = keyboard_check(vk_right) or keyboard_check(ord("D"));  
var up = keyboard_check(vk_up) or keyboard_check(ord("W"));  
var down = keyboard_check(vk_down) or keyboard_check(ord("S"));  
  
var hor = (right - left);  
var vertical = (down - up);
```

Hình 3.6 Lệnh di chuyển

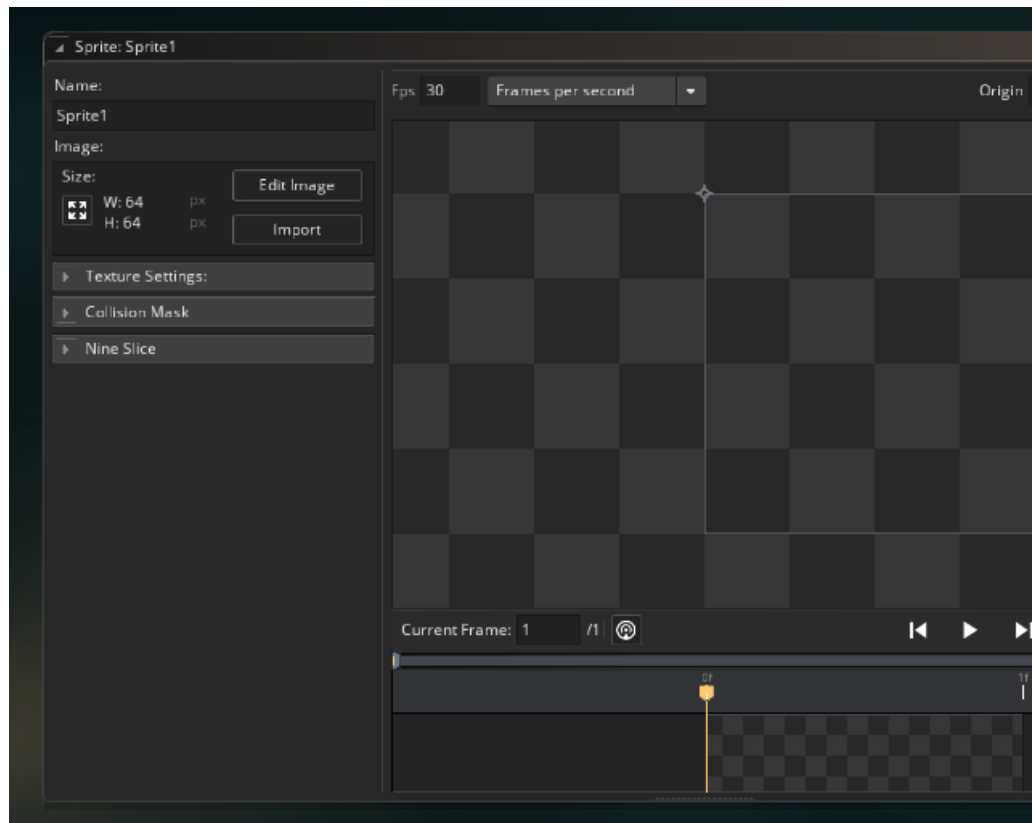
Để nhân vật có được hoạt ảnh cũng như giao diện bên ngoài, cần gán cho Object của nhân vật một Sprite.

Tạo Sprite: chuột phải vào Sprite, chọn Create, Sprite



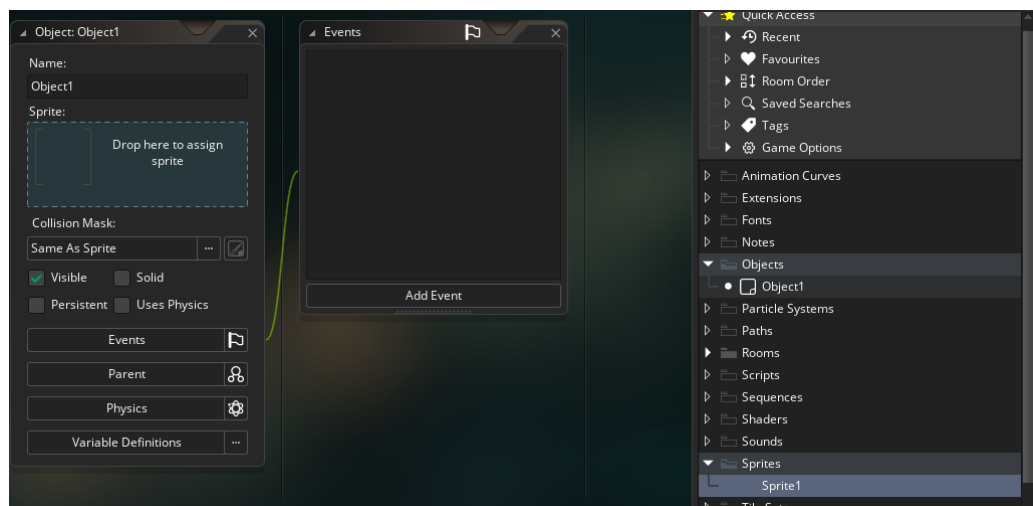
Hình 3.7 Tạo Sprite

Tại Sprite vừa tạo chọn Import để thêm hình ảnh có sẵn hoặc Edit Images để vẽ trực tiếp trên GameMaker



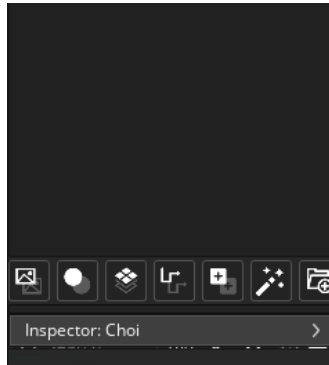
Hình 3.8 Thêm hình ảnh vào Sprite

Sau khi tạo xong một Sprite hoàn chỉnh, kéo Sprite đó vào Object nhân vật để gán hoạt ảnh cho nhân vật.

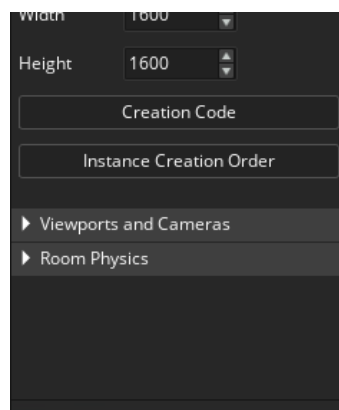


Hình 3.9 Gán Sprite cho Object

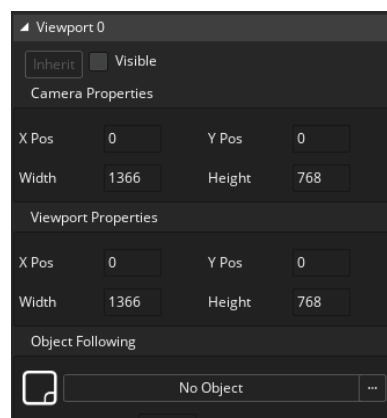
Để Camera lúc nào cũng di chuyển theo nhân vật Vào Room Chơi.
Chọn Inspector.



Hình 3.10 Camera theo nhân vật
Chọn viewports and Cameras.



Hình 3.11 Camera theo nhân vật
Tại Viewport 0 kéo Object Nhân vật vào



Hình 3.12 Camera theo nhân vật

3.3. Tạo map chơi, giao diện và cơ chế cho Menu, Chọn nhân vật

3.3.1 Tạo map chơi

Để tạo một map chơi cần có các Tileset

Tileset: là một mạng lưới các **gạch** được sử dụng để tạo ra map (bản đồ) cho game.

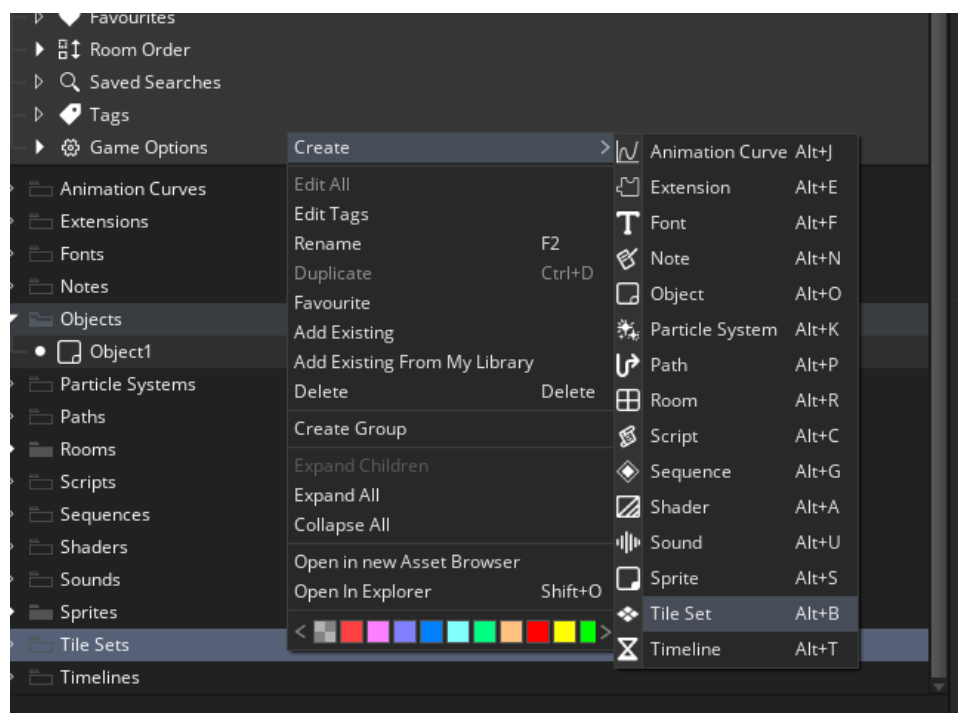


Hình 3.13 Tileset

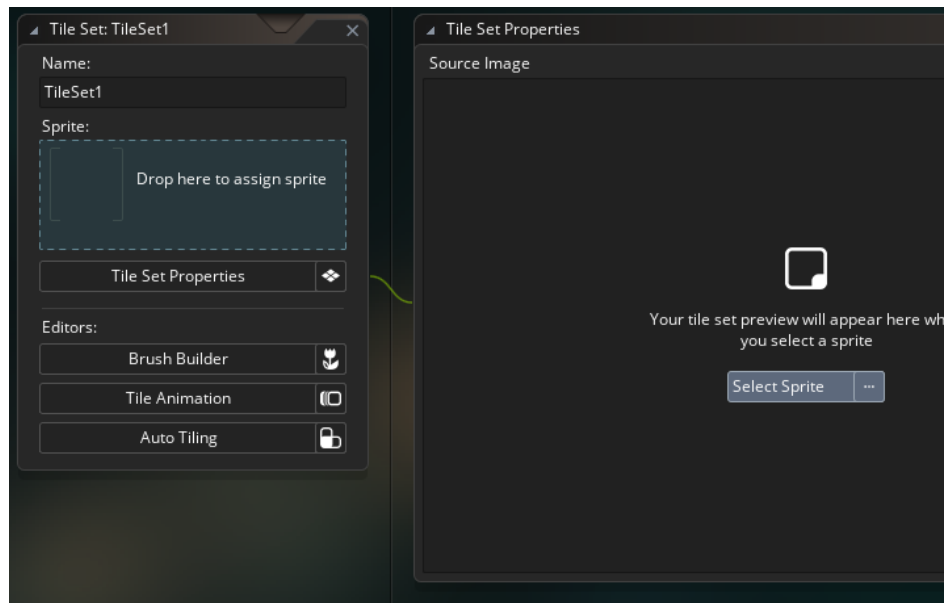
Để thêm Tileset ta làm giống như tạo Object và thêm hoạt ảnh cho Object.

Bấm chuột phải vào Tileset chọn Create, Tileset.

Rồi kéo thả Sprite của Tileset vào.

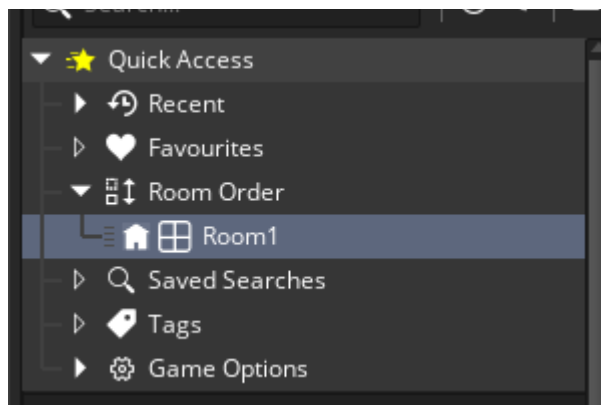


Hình 3.14 Tạo Tileset



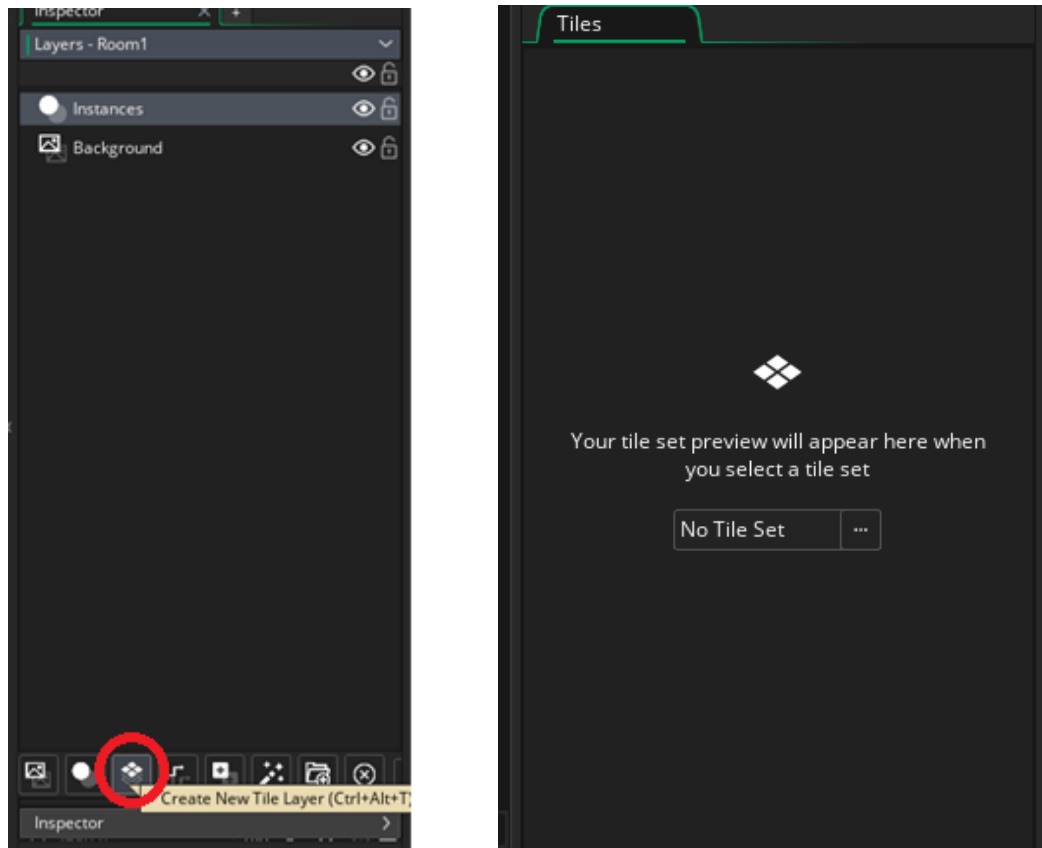
Hình 3.15 Thêm Sprite vào Tileset

Sau khi tạo Tileset, mở giao diện Room bằng cách nhấp đúp chuột vào Room



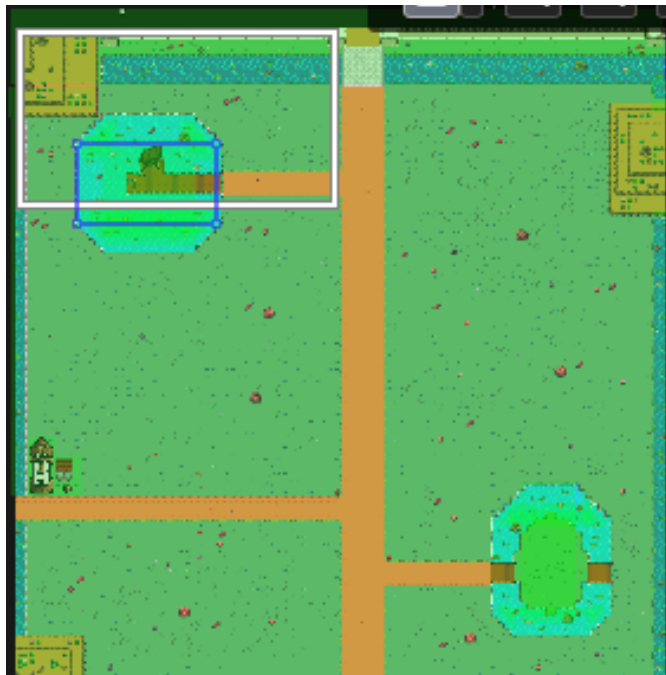
Hình 3.16 Thêm Tileset vào Room

Ở mục Inspector bên trái màn hình chọn Create New Tile Layer, rồi thêm Tileset của mình vào.



Hình 3.17 Thêm Tileset vào Room

Sau đó dung những Tileset đó vẽ thành máp hoàn chỉnh

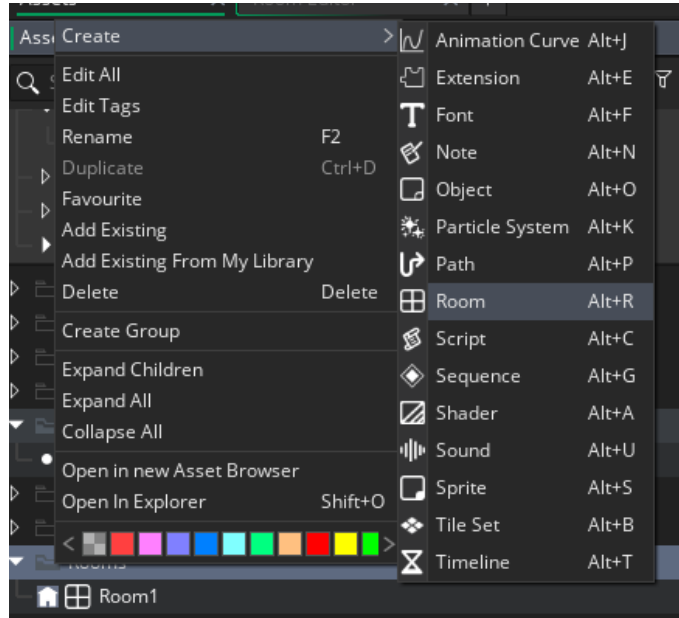


Hình 3.18 Map Chơ

3.3.2 Tạo giao diện và cơ chế cho Menu

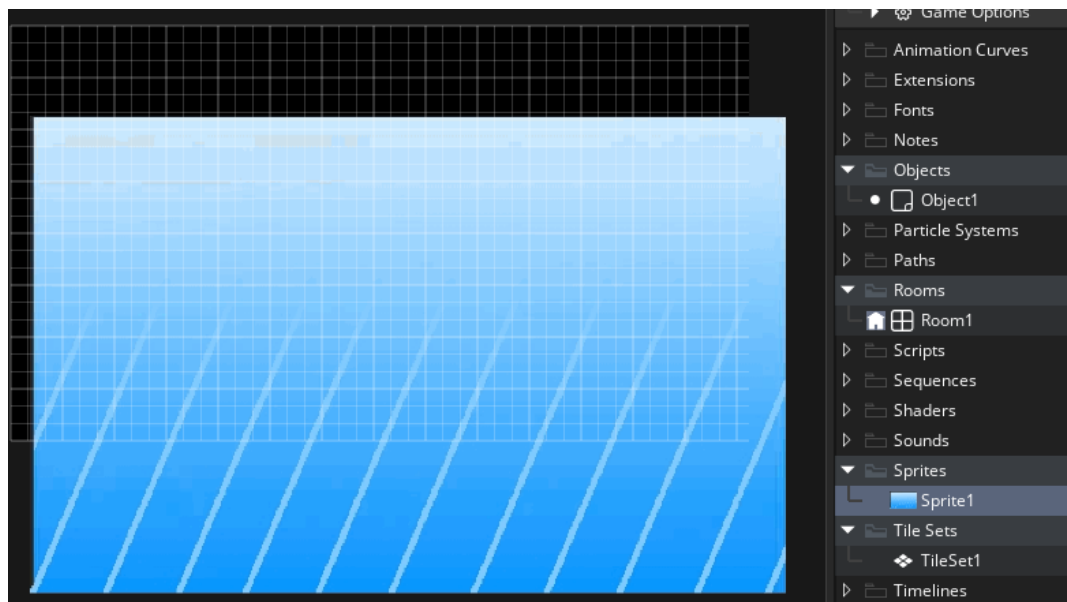
Để tạo giao diện Menu, cần tạo một Room mới.

Nhấn chuột phải vào Room chọn Create, Room



Hình 3.19 Tạo Room

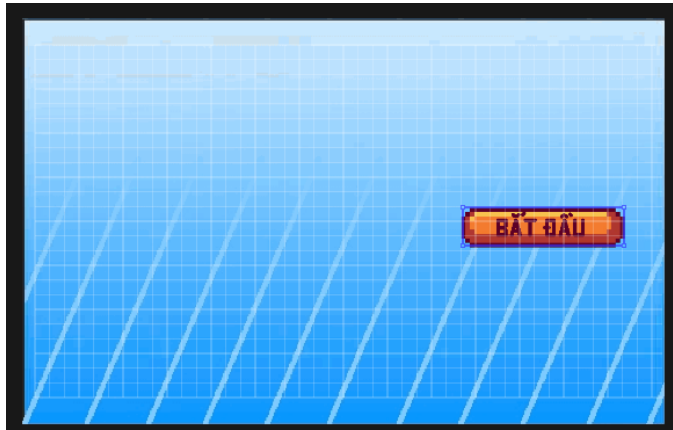
Tạo Background cho Room Menu bằng cách kéo Sprite cho Background vào Room



Hình 3.20 Thêm Background cho Room

Sau khi tạo Background cho Room Menu tiếp tục tạo các nút điều hướng dùng để di chuyển qua các Room khác như: Bắt đầu, nâng cấp, tùy chỉnh, Thoát.

Để tạo nút bắt đầu ta tạo Object cho nút bắt đầu dùng Sprite phù hợp, rồi kéo vào Room Menu chọn vị trí phù hợp.



Hình 3.21 Thêm nút điều hướng vào Room

Làm tương tự các nút còn lại



Hình 3.22 Thêm các nút điều hướng vào Room

Để khi nhấn vào các nút có thể điều hướng ta cần tạo các Room tương ứng và lập trình cho các nút.

Vào Object của nút thêm Event Left Pressed, rồi thêm lệnh vào

Sử dụng lệnh `room_goto(tên Room muốn chuyển)` để chuyển qua Room đó.

```
room_goto(Chonnhavai);
```

Hình 3.23 Lệnh chuyển Room

Đối với nút Thoát dùng lệnh `game_end()` để thoát game.

Có thể thêm các sự kiện Mouse Enter để khi chuột nằm bên trong nút sẽ to ra.

```
image_xscale = 1.1;  
image_yscale = 1.1;
```

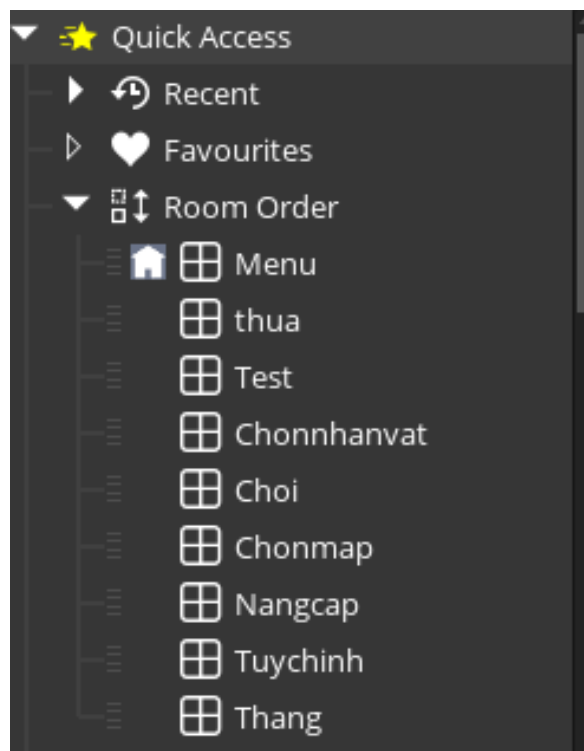
Hình 3.24 Lệnh phóng to nút

Sự kiện Mouse Leave để khi chuột ở ngoài nút sẽ nhỏ lại.

```
image_xscale = 1;  
image_yscale = 1;
```

Hình 3.25 Lệnh thu nhỏ nút

Muốn khi Game được mở lên Room đầu tiên là Menu. Tại Room Order, kéo Room muốn mở đầu tiên lên trên cùng.



Hình 3.26 Đổi Room đầu tiên

3.3.3 Tạo giao diện và cơ chế chọn nhân vật

Vào Room Chonnhanvat Thêm Background tương tự Room Menu

Tạo Các Object thẻ chọn nhân vật rồi bỏ vào Room



Hình 3.27 Menu chọn nhân vật

Tạo thêm Sprite của thẻ chọn nhân vật để biểu thị là đang chọn nhân vật đó



Hình 3.28 Thẻ chọn nhân vật

Để khi chọn vào Object thẻ tương ứng có thể tạo ra nhân vật mình chọn, cần thêm vào Event Left Pressed cho từng Object thẻ.

Global. dùng để khi đổi Room thì biến sau nó vẫn giữ được giá trị

Thêm **sprite_index = (Sprite của thẻ được chọn)** để khi người chơi bấm vào Object của thẻ này thì Sprite hiện tại sẽ đổi sang Sprite được chọn. Làm tương tự với các Object thẻ khác.

```
global.nhanvat_duocchon = 1;  
  
sprite_index = spr_chonnhanvat1_1;  
  
with (obj_chonnhanvat2) {  
    sprite_index = spr_chonnhanvat2;  
}  
with (obj_chonnhanvat3) {  
    sprite_index = spr_chonnhanvat3;  
}
```

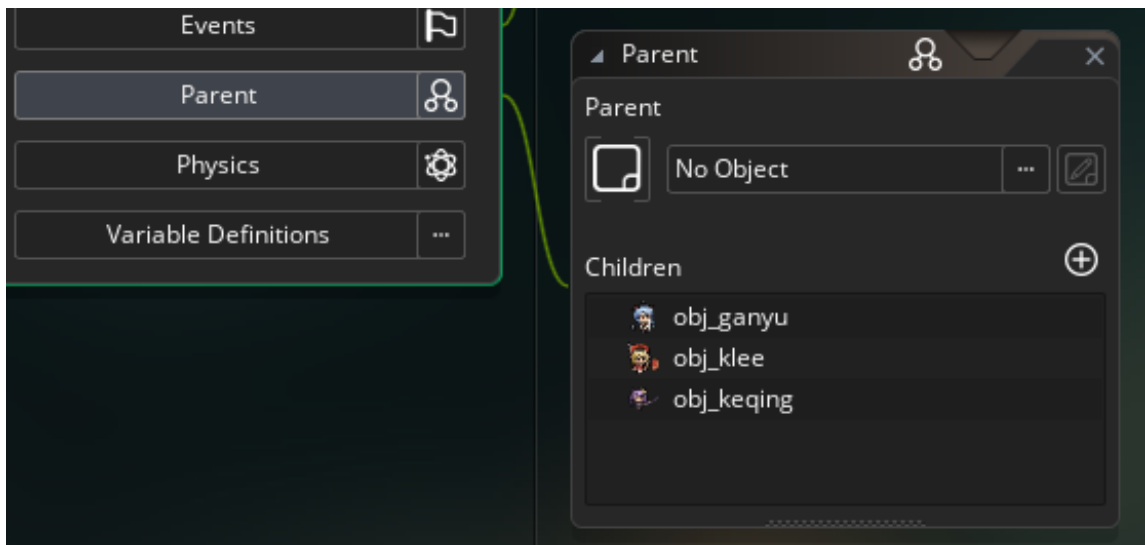
Hình 3.29 Lệnh chọn nhân vật

Sau đó quay lại Room chơi, tạo và thêm Object Controller vào Room Chơi

Object này sẽ dùng để tạo ra nhân vật mà mình chọn.

Tạo thêm Object cho đủ các nhân vật

Mở lại Object nhân vật ban đầu thêm các Object nhân vật vừa tạo vào Parent của nhân vật ban đầu. Bằng cách này chỉ cần lập trình cho nhân vật ban đầu thì tất cả các nhân vật vừa tạo sẽ được thừa hưởng toàn bộ lệnh của nhân vật ban đầu.



Hình 3.30 Thêm các nhân vật vào Parent

Quay lại Object Controller để lập trình tạo nhân vật được chọn. Sử dụng Event Create để tạo nhân vật được chọn.

```
if (global.nhanvat_duocchon == 1) {  
    instance_create_layer(850, 800, "Instances", obj_keqing);}  
if (global.nhanvat_duocchon == 2) {  
    instance_create_layer(850, 800, "Instances", obj_klee);}  
if (global.nhanvat_duocchon == 3) {  
    instance_create_layer(850, 800, "Instances", obj_ganyu);}
```

Hình 3.31 Lệnh tạo nhân vật được chọn

Về Room Chọn nhân vật thêm các nút trở lại để về Room Menu, Nút chơi để vào Room chơi.



Hình 3.32 Room chọn nhân vật

3.4. Tạo quái, boss và cơ chế tấn công của nhân vật

3.4.1 Tạo quái và boss

Tạo một Object Quái, cho các Object quái 1, quái 2 ,quái 3,... vào Parent của Object quái

Tương tự với Boss.

Để Quái và Boss có thể đi theo nhân vật, cần thêm vào Event Step của Quái và Boss obj_noelle là Object nhân vật ban đầu, walkSpeed là tốc độ di chuyển của quái có thể thay bằng một giá trị số thực nào đó.

```
with(obj_quai){  
    image_speed = 1;  
    direction = point_direction(x, y, obj_noelle.x, obj_noelle.y);  
    speed = walkSpeed;
```

Hình 3.33 Lệnh quái theo nhân vật

3.4.2 Cơ chế tấn công

Thêm các giá trị như hp cho Quái và Boss hp, damage_delay = 0, damage_cooldown = 30 cho nhân vật vào Event Create. Với damage_delay và damage_cooldown dùng để giới hạn số lần gây sát thương của một lần chạm dành cho nhân vật tránh tình trạng nhân vật chánh tình trạng chạm một phát hp nhân vật về 0.

Cơ chế tấn công của quái khi Quái và Boss chạm vào gây sát thương cho nhân vật. Thêm vào Event Step của nhân vật.

```
if (place_meeting(x, y, obj_quai) || place_meeting(x, y, obj_boss)) {  
    if (damage_delay >= damage_cooldown) {  
        hp -= 10;  
        damage_delay = 0;  
    } else {  
        damage_delay += 1;  
    }  
}
```

Hình 3.34 Lệnh quái gây sát thương

Cơ chế tấn công của nhân vật, tạo một Object kỹ năng làm Parent của các kỹ năng 1, kỹ năng 2,...

Đoạn lệnh dưới được gọi là **ds_list_create()** là một hàm phổ biến với việc lập trình cơ chế trong GameMaker. Dùng để lấy chính xác id đang được nhắc đến, khi có nhiều Object cùng loại được tạo ra cùng lúc.

```
var _list = ds_list_create();  
var num = collision_rectangle_list(bbox_left, bbox_top, bbox_right, bbox_bottom, obj_quai, false, true, _list, false);  
if (_num > 0)  
{  
    for (var i = 0; i < _num; ++i)  
    {  
        if (list[i].sprite_index == list[i].spr){  
            var dir = point_direction(x, y, _list[i].x, _list[i].y)  
            _list[i].x = _list[i].x + lengthdir_x(5, dir);  
            _list[i].y = _list[i].y + lengthdir_y(5, dir);  
  
            list[i].hp = dmg;  
            if (_list[i].hp <= 0){  
                instance_destroy(_list[i]);  
            }  
        }  
    }  
}  
ds_list_destroy(_list);
```

Hình 3.35. Lệnh gây sát thương cho quái

Tại Event Create của nhân vật tạo thêm các Alarm dùng để đếm thời gian cho các kỹ năng của nhân vật. Các dmg để lưu lượng sát thương gây ra của các kỹ năng.

```
alarmtime[0] = 80;  
alarmtime[1] = 120;  
alarmtime[2] = 120;  
alarmtime[3] = 360;  
alarmtime[4] = 480;  
  
alarm[0] = alarmtime[0];  
alarm[1] = alarmtime[1];  
alarm[2] = alarmtime[2];  
alarm[3] = alarmtime[3];  
alarm[4] = alarmtime[4];
```

```
dmg_congthem = 0;  
dmg_trongkiem = 2 + global.dmgngang + dmg_congthem;  
dmg_sach = 3 + global.dmgngang + dmg_congthem;  
dmg_ten = 2 + global.dmgngang + dmg_congthem;  
dmg_banhxlua = 2 + global.dmgngang + dmg_congthem;  
dmg_boomnay = 2 + global.dmgngang + dmg_congthem;
```

Hình 3.36 Các biến Alarm

Tạo Event Alarm tương ứng để khi đếm thời gian của các kỹ năng về 0 thì kỹ năng đó được kích hoạt. Lập trình thêm khi Alarm đó về 0 thì điều gì xảy ra. Có thể thêm global.nhanvat_duocchon để giới hạn các nhân vật khác không dùng được kỹ năng này.

```
if (global.nhanvat_duocchon == 1)  
{  
    var trongkiem = instance_create_layer(x,y, "Instances", obj_trongkiem);  
    trongkiem.image_xscale = image_xscale;  
  
    if (image_xscale == 1){ trongkiem.addX= 5; }  
    if (image_xscale == -1){ trongkiem.addX= -5; }  
}
```

Hình 3.37 Lệnh kỹ năng

Làm tương tự với các kỹ năng khác và kỹ năng của Boss.

3.5. Quái xuất hiện ngẫu nhiên và bộ đếm thời gian

3.5.1 Quái xuất hiện

Tạo các Event Alarm các biến thời gian cho các Alarm đó để gọi Quái.

Trong Alarm dùng spawn_point để lưu các điểm xuất hiện.


```
var spawn_points = [
  [173, 742],
  [265, 1496],
  [680, 1561],
  [1084, 1445],
  [607, 602],
  [1327, 301],
  [664, 204],
  [939, 629],
  [1174, 918],
  [754, 1238]
];
```

Hình 3.38 Biến lưu vị trí xuất hiện của quái

Dùng hàm **irandom** để chọn ngẫu nhiên một điểm xuất hiện và triệu hồi Quái ở điểm đó.

```
repeat(1) {
  // Lựa chọn điểm spawn ngẫu nhiên từ danh sách
  var random_spawn = spawn_points[irandom(array_length(spawn_points) - 1)];
  var spawn_x = random_spawn[0];
  var spawn_y = random_spawn[1];

  // Kiểm tra xem điểm spawn có nằm trong phạm vi camera không
  var _vx = camera_get_view_x(view_camera[0]);
  var _vy = camera_get_view_y(view_camera[0]);
  var _vw = camera_get_view_width(view_camera[0]);
  var _vh = camera_get_view_height(view_camera[0]);

  if (spawn_x < _vx || spawn_x > (_vx + _vw) || spawn_y < _vy || spawn_y > (_vy + _vh)) {
    // Nếu điểm spawn nằm ngoài phạm vi camera, tạo quái vật tại đó
    instance_create_layer(spawn_x, spawn_y, "Instances", obj_hilichult_1);
  } else {
    // Nếu điểm spawn nằm trong phạm vi camera, thử lại
    alarm[1] = alaTime1;
  }
}
```

Hình 3.39 Lệnh cho quái xuất hiện ngẫu nhiên

Làm tương tự với các con Quái khác và Boos

3.5.2 Bộ đếm thời gian

Tạo một Object Đồng hồ

Khởi tạo các giá trị trong Event Create. Sử dụng global để ghi nhớ khoảng thời gian để khi thua hoặc thắng có thể in thời gian chơi ra.

```
formatted_minutes = 0;
formatted_seconds = 0;

giay = 0;
phut = 0;
count_up = false;

global.phutdachoi = 0;
global.giaydachoi = 0;
```

Hình 3.40 Các biến của bộ đếm thời gian

Lập trình cho Object đồng hồ cơ chế thời gian với **room_speed** là số frame hình trong một giây.

```
if (instance_exists(obj_lencap)) {
    count_up = false;
}
else
{
    count_up = true;
}

if (count_up == true) {
    giay += 1/room_speed;
}

if (giay < 60) && (giay > 59.9)
{
    giay = 0;
    phut += 1;
}

// Đảm bảo phút và giây luôn có 2 chữ số
formatted_minutes = (phut < 10) ? "0" + string(phut) : string(phut);
formatted_seconds = (giay < 10) ? "0" + string(giay) : string(giay);
```

Hình 3.41 Lệnh của bộ đếm thời gian

Để in được thời gian trong lúc chơi ta sử dụng Event Draw để có thể cập liên tục thời gian trong lúc chơi.

```
// Lấy tọa độ của camera
var camera_x = camera_get_view_x(view_camera[0]) + 390;
var camera_y = camera_get_view_y(view_camera[0]) + 10 ;

x = camera_x + 270;
y = camera_y + 34;

// Hiển thị đồng hồ
draw_set_color(c_white);
draw_text(camera_x, camera_y, formatted_minutes + ":" + formatted_seconds);
```

Hình 3.42 Lệnh vẽ bộ đếm thời gian

3.6. Cơ chế lên cấp và hệ thống nâng cấp kỹ năng

3.6.1 Cơ chế lên cấp

Tạo một Object là Kinh nghiệm khai báo `canFly = False` trong Event Create. Lập trình cho Event Step khi nhân vật ở một khoảng cách nhất định thì kinh nghiệm sẽ bay về nhân vật, tạo hiệu ứng nhật kinh nghiệm.

```
if(canFly == true){  
    move_towards_point(obj_noelle.x, obj_noelle.y, 2);  
}
```

Hình 3.43 Lệnh nhật kinh nghiệm

Để khi nhật được số kinh nghiệm nhất định thì nhân vật lên cấp cần khai báo thêm 2 biến `xp = 0` và `xpNext = 10` cho Object với `xp` là giá trị kinh nghiệm đang có và `xpNext` là số kinh nghiệm cần để lên cấp. Sau đó thêm Event Collision giữa nhân vật và kinh nghiệm để khi 2 thứ đó chạm vào nhau sẽ thực hiện lệnh.

```
xp += 1;  
if (xp >= xpnext){  
    xp = 0;
```

Hình 3.44 Lệnh lên cấp

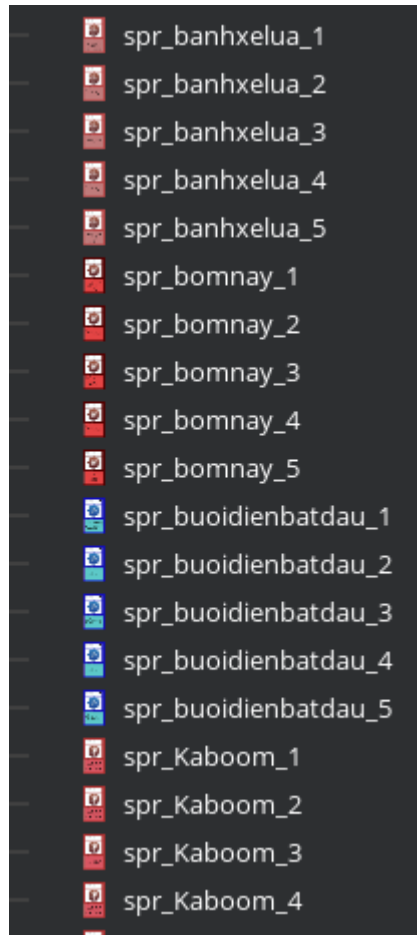
Để hiện thị thanh kinh nghiệm ta thêm vào Event Draw của nhân vật

```
draw_set_color(c_black);  
draw_rectangle(x - bar_width / 2, y - bar_height / 2, x + bar_width / 2, y + bar_height / 2, false);  
  
var xp_ratio = xp / xpnext;  
draw_set_color(c_blue);  
draw_rectangle(x - bar_width / 2 + 1, y - bar_height / 2 + 1, x - bar_width / 2 + (bar_width - 2) * xp_ratio, y + bar_height / 2 - 1, false);
```

Hình 3.45 Lệnh vẽ thanh kinh nghiệm

3.6.2 Hệ thống nâng cấp kỹ năng

Tạo các Sprite tương ứng với các kỹ năng và cấp độ mong muốn



Hình 3.46 Sprite của các kỹ năng

Tại Event Collision giữa nhân vật và kinh nghiệm bên trong lệnh nâng cấp khai báo item_sprites lưu các sprite kỹ năng lại.

```
var item_sprites = [  
    [spr_vanlai_1, spr_vanlai_2, spr_vanlai_3, spr_vanlai_4, spr_vanlai_5],  
    [spr_buoidienbatdau_1, spr_buoidienbatdau_2, spr_buoidienbatdau_3, spr_buoidienbatdau_4, spr_buoidienbatdau_5],  
    [spr_bomnay_1, spr_bomnay_2, spr_bomnay_3, spr_bomnay_4, spr_bomnay_5],  
    [spr_tanghp_1, spr_tanghp_2, spr_tanghp_3, spr_tanghp_4, spr_tanghp_5],  
    [spr_tangdame_1, spr_tangdame_2, spr_tangdame_3, spr_tangdame_4, spr_tangdame_5],  
    [spr_tangtoc_1, spr_tangtoc_2, spr_tangtoc_3, spr_tangtoc_4, spr_tangtoc_5],  
    [spr_banhxlua_1, spr_banhxlua_2, spr_banhxlua_3, spr_banhxlua_4, spr_banhxlua_5],  
    [spr_themora]  
];
```

Hình 3.47 Biến lưu các Sprite của kỹ năng

Khai báo thêm một vài biến để lưu giá trị cho việc chọn chọn ngẫu nhiên 3 thẻ với:

SoNgauNhiem: tương ứng với các hàng trong **item_sprites**

SoVatPhamDaChon: là số kỹ năng khác nhau mà người chơi đã chọn

Vp_1, vp_2,...: là cấp độ của của các kỹ năng

```
SoNgauNhiem = [0, 1, 2, 3, 4, 5, 6];  
SoVatPhamDaChon = 0;  
CacVatPhamDaChon = [];  
vp_1=0; vp_2=0; vp_3=0; vp_4=0; vp_5=0; vp_6=0; vp_7=0;  
VatPhamConLai = 5;
```

Hình 3.48 Biến cho hàm chọn ngẫu nhiên

Tạo hàm ngẫu nhiên để chọn ngẫu nhiên 3 số không trùng nhau tương ứng với 3 hàng của **item_sprites** rồi lưu nó vào mảng **SoDuocRandomRa**.

```
var SoDuocRandomRa = [];  
if(SoVatPhamDaChon < 5)  
{  
  while (array_length(SoDuocRandomRa) < 3)  
  {  
    var index = irandom(array_length(SoNgauNhiem) - 1); // Chọn một chỉ số ngẫu nhiên trong mảng arr  
    var selected_item = SoNgauNhiem[index]; // Lấy phần tử tại chỉ số ngẫu nhiên  
    // Kiểm tra xem phần tử đã được chọn chưa  
    if (!array_contains(SoDuocRandomRa, selected_item))  
    {  
      array_push(SoDuocRandomRa, selected_item); // Thêm phần tử vào mảng chọn nếu chưa có  
    }  
  }  
}  
else  
{  
  if(VatPhamConLai == 3)  
  {  
    array_push(CacVatPhamDaChon, 7);  
  }  
  if(VatPhamConLai == 2)  
  {  
    array_push(CacVatPhamDaChon, 8);  
  }  
  if(VatPhamConLai == 1)  
  {  
    array_push(CacVatPhamDaChon, 9);  
  }  
  
  while (array_length(SoDuocRandomRa) < 3)  
  {  
    var index = irandom(array_length(CacVatPhamDaChon) - 1);  
    var selected_item = CacVatPhamDaChon[index];  
    if (!array_contains(SoDuocRandomRa, selected_item))  
    {  
      array_push(SoDuocRandomRa, selected_item);  
    }  
  }  
}
```

Hình 3.49 Lệnh chọn ra 3 số ngẫu nhiên

Sau đó dựa vào 3 số ngẫu nhiên để in ra màn hình các sprite tương ứng.

```
for (var i = 0; i < 3; i++) {
    if(SoDuocRandomRa[i] == 0)
    {
        var _sprite_index = item_sprites[SoDuocRandomRa[i]][vp_1];
        var xPos = _vx + (i - 1) * 250; // Khoảng cách giữa các thẻ
        var obj = instance_create_depth(xPos, _vy, -2000, obj_lencap);
        obj.destinationx = xPos - (i - 1) * 60;
        obj.destinationy = _vy;
        obj.image_alpha = 1;
        obj.sprite_index = _sprite_index;
    }
    if(SoDuocRandomRa[i] == 1)
    {
        var _sprite_index = item_sprites[SoDuocRandomRa[i]][vp_2];
        var xPos = _vx + (i - 1) * 250; // Khoảng cách giữa các thẻ
        var obj = instance_create_depth(xPos, _vy, -2000, obj_lencap);
        obj.destinationx = xPos - (i - 1) * 60;
        obj.destinationy = _vy;
        obj.image_alpha = 1;
        obj.sprite_index = _sprite_index;
    }
}
```

Hình 3.50 Lệnh in ra 3 Sprite

Tiếp theo tạo một Object Lên cấp và bỏ vào Room Chơi. Object này sẽ xử lý việc khi người chơi chọn vào một trong 3 Sprite sẽ thực hiện điều tương ứng với thẻ đó bằng Event Left Released.

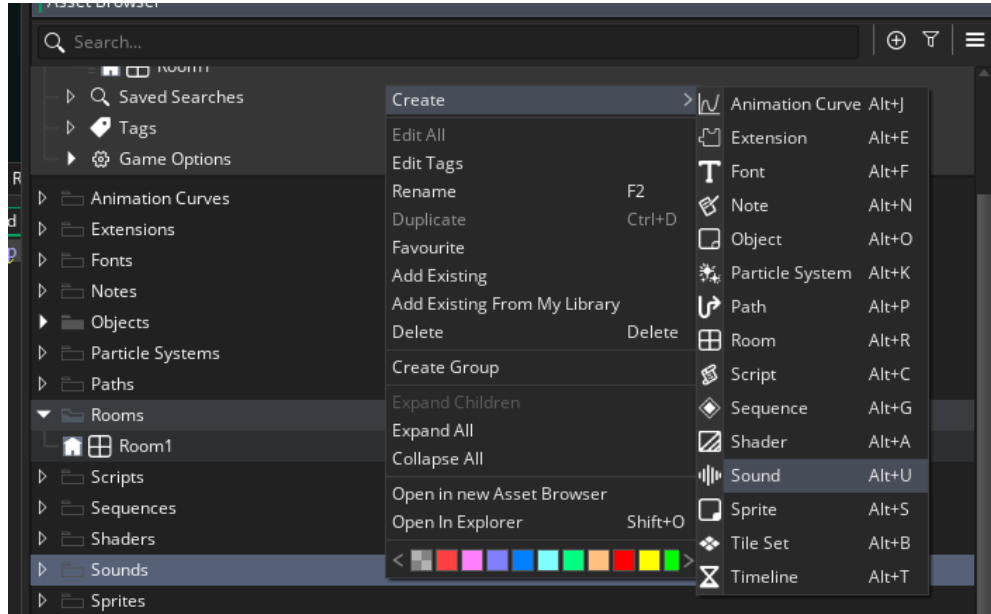
```
if (sprite_index == spr_vanlai_1)
{
    obj_noelle.vp_1 ++;
    obj_noelle.dmg_trongkiem ++;
    obj_noelle.SoVatPhamDaChon ++;
    array_push(obj_noelle.CacVatPhamDaChon, 0);
}
if(sprite_index == spr_vanlai_2 || sprite_index == spr_vanlai_3 || sprite_index == spr_vanlai_4 || sprite_index == spr_vanlai_5)
{
    obj_noelle.dmg_trongkiem ++;
    obj_noelle.vp_1 ++;
    if(obj_noelle.vp_1 == 5)
    {
        obj_noelle.VatPhamConLai --;
        var index = array_index_of(obj_noelle.CacVatPhamDaChon, 0);
        var index1 = array_index_of(obj_noelle.SoNgauNhien, 0);
        if (index != -1) {
            array_delete(obj_noelle.CacVatPhamDaChon, index,1);
        }
        if (index1 != -1)
        {
            array_delete(obj_noelle.SoNgauNhien, index1,1);
        }
    }
}
}
```

Hình 3.51 Lệnh nâng cấp các kỹ năng

3.7. Thêm nhạc nền, âm thanh hiệu ứng và giao diện tùy chỉnh

3.7.1 Thêm nhạc nền và âm thanh hiệu ứng

Thêm Sound, chuột phải vào Sound, chọn Create, Sound



Hình 3.52 Tạo Sound

Tạo một Object Nhạc Menu để gọi các Sound lên. Với `music_volume` là âm lượng với giá trị mặc định là 1.

```
// Kiểm tra các room có nhạc nền chung
if room == Menu || room == Tuychinh || room == Nangcap || room == Chonmap || room == Chonnhavtat || room == th
{
    if global.menu_music == noone {
        global.menu_music = snd_menu;
        audio_play_sound(snd_menu, global.music_volume, true);
        audio_sound_gain(global.menu_music, global.music_volume, 0);
    }
} else {
    if global.menu_music == snd_menu {
        audio_stop_sound(global.menu_music);
        global.menu_music = noone;
    }
}

if room == Choi {
    if global.menu_music == noone {
        global.menu_music = snd_choi;
        audio_play_sound(snd_choi, global.music_volume, true);
        audio_sound_gain(global.menu_music, global.music_volume, 0);
    }
} else {
    if global.menu_music == snd_choi {
        audio_stop_sound(global.menu_music);
        global.menu_music = noone;
    }
}
```

Hình 3.53 Lệnh phát nhạc nền

Làm tương tự với hiệu ứng âm thanh.

3.7.2 Giao diện và hệ thống tùy chỉnh

Trong Room Tùy chỉnh thêm vào các nút tăng giảm âm lượng với Event Left Pressed để tăng giảm âm lượng.

```
if(room == Tuychinh)
{
    global.music_volume = min(global.music_volume + 0.1, 1);
    audio_sound_gain(global.menu_music, global.music_volume, 0);
}
```

Hình 3.54 Lệnh tăng âm lượng nhạc nền

Để hiển thị thanh âm lượng thêm vào Event Draw của Nhạc Menu

```
draw_self()
var music_width = 250;
var music_height = 20;

var music_length = music_width * global.music_volume;
draw_set_font(font_vietnamese);
draw_set_color(c_black);
draw_text(x - music_width - 10, y, "Nhạc nền: ");

draw_set_color(c_gray);
draw_rectangle(x, y, x + music_width, y + music_height, false);

draw_set_color(c_green);
draw_rectangle(x, y, x + music_length, y + music_height, false);
```

Hình 3.55 Lệnh hiển thị nhạc nền

Làm tương tự với hiệu ứng âm thanh.

3.8. Hệ thống nâng cấp bằng tiền

Vào Room Nâng cấp Thêm vào các nút nâng dame, nâng tốc, nâng máu. Rồi thêm vào Event Left Pressed.


```
if (global.tongtien >= global.upgrade_cost_damage) {  
    global.tongtien -= global.upgrade_cost_damage;  
    global.dmgngang += 0.2; // Tăng damage  
    global.upgrade_cost_damage += 10; // Giá tăng  
    txt = "Nâng cấp Damage: " + string(global.upgrade_cost_damage) + " tiền";  
} else {  
    show_message("Bạn không đủ tiền!");  
}
```

Hình 3.56 Lệnh nút nâng cấp

Thêm vào Event Draw để hiển thị các dòng txt.

```
draw_self();  
draw_set_font(font_vietnamese);  
draw_set_color(c_black);  
draw_text(x + 40, y - 10, txt);
```

Hình 3.57 Lệnh in dòng txt

Để hiển thị các chỉ số đã được cộng thêm và số tiền hiện có, tạo thêm một Object Hiển thị và Event Draw.

```
draw_text(10, 10, "Mora đang có: " + string(global.tongtien));  
draw_text(10, 30, "Damage cộng thêm: " + string(global.dmgngang));  
draw_text(10, 50, "Speed cộng thêm: " + string(global.spdngang));  
draw_text(10, 70, "HP cộng thêm: " + string(global.hpngang));
```

Hình 3.58 Lệnh hiển thị thông số và tiền hiện có

Làm tương tự với các nút còn lại.

CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU

4.1. Tổng quan kết quả

Đồ án đã hoàn thành các mục tiêu đề ra ban đầu. Sản phẩm cuối cùng là một trò chơi có thể chơi được, với đầy đủ các chức năng cơ bản và trải nghiệm mượt mà. Trò chơi đã được xuất ra dưới định dạng .exe, đảm bảo khả năng chạy ổn định trên nền tảng Windows.

Link GitHub của đồ án: <https://github.com/tranlamphuduc/csn-da22ttd-tranlamphuduc-lamgameroguelike-gamemaker.git>

Link tải game: <https://drive.google.com/file/d/1WF9XSJkiN--jkH83V1UYAV9XBL7qgvpo/view?usp=sharing>

4.2. Kết quả đạt được

4.2.1 Giao diện Menu chính

Menu chính được thiết kế đơn giản nhưng trực quan, với các tùy chọn có thể tương tác được.



Hình 4.1 Giao diện Menu chính

4.2.2 Cơ chế và giao diện chọn nhân vật

Giao diện với ba nhân vật được biểu hiện bởi ảnh đại diện tương ứng với nhân vật đó, cùng với các thông số và kỹ năng khác nhau. Người chơi có thể chọn nhân vật mình thích để bắt đầu chơi.



Hình 4.2 Giao diện chọn nhân vật

4.2.3 Cơ chế và giao diện nâng cấp

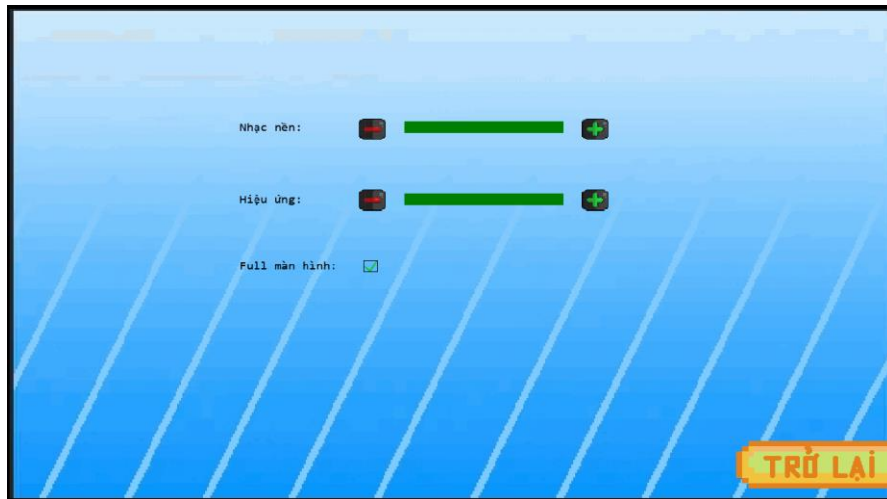
Các hiển thị về thông số đã mua cũng với mora (tiền trong game) nằm bên trái và các nút nâng cấp.



Hình 4.3 Giao diện nâng cấp

4.2.4 Cơ chế và giao diện tùy chỉnh

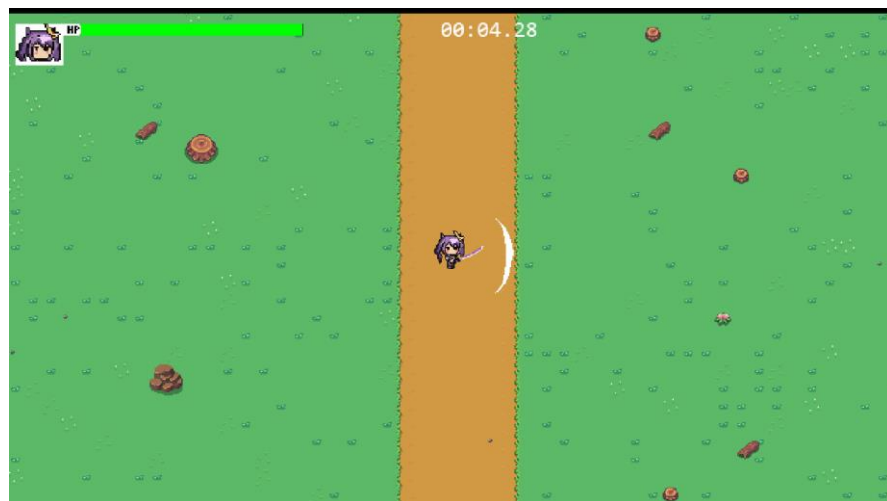
Giao diện tùy chỉnh với 3 chức năng chính tăng giảm âm lượng nhạc nền, tăng giảm âm lượng hiệu ứng, bật tắt toàn màn hình.



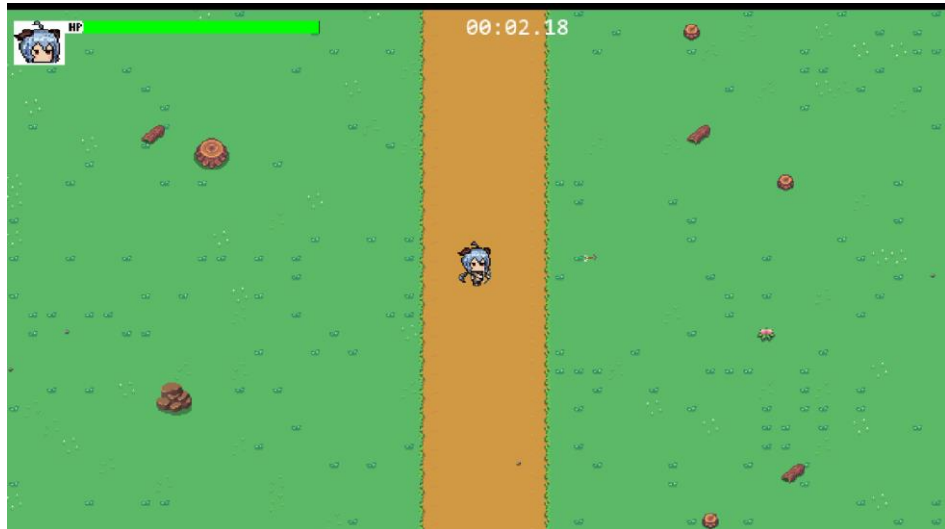
Hình 4.4 Giao diện tùy chỉnh

4.2.5 Cơ chế chiến đấu

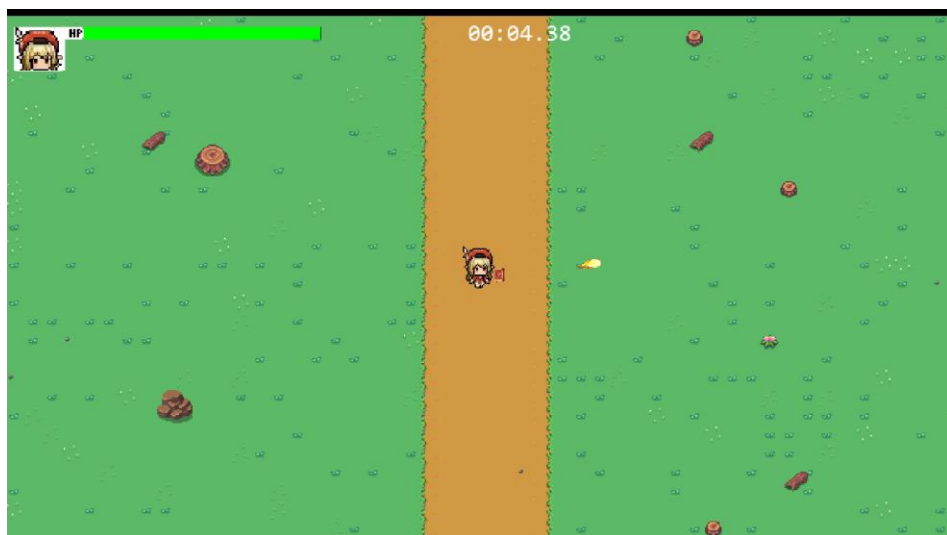
Nhân vật sẽ tự động tấn công kẻ thù theo hướng chỉ định của người chơi, người chơi có thể điều khiển nhân vật né tránh và sử dụng kỹ năng đặc biệt để hỗ trợ tiêu diệt kẻ địch. Mỗi nhân vật có 1 kỹ năng riêng và 6 kỹ năng chung sẽ được mở khóa hoặc nâng cấp khi nhân vật lên cấp.



Hình 4.5 Kỹ năng riêng của nhân vật 1



Hình 4.6 Kỹ năng riêng của nhân vật 2



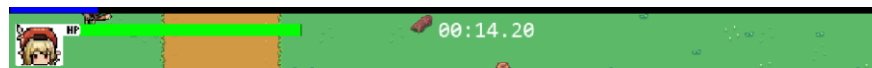
Hình 4.7 Kỹ năng riêng của nhân vật 3

4.2.6 Cơ chế lên cấp và chọn kỹ năng

Khi tiêu diệt kẻ thù, nhân vật sẽ thu thập điểm kinh nghiệm để lên cấp. Khi lên cấp, người chơi nhận được các thẻ vật phẩm để nhân thêm kỹ năng hoặc nâng cấp các kỹ năng hiện có. Các cấp độ cao hơn sẽ yêu cầu nhiều điểm kinh nghiệm hơn.



Hình 4.8 Cơ chế rút kinh nghiệm khi tiêu diệt quái



Hình 4.9 Thanh kinh nghiệm



Hình 4.10 Cơ chế lên cấp và chọn kỹ năng

4.2.7 Kẻ thù xuất hiện theo thời gian

Kẻ thù sẽ xuất hiện theo thời gian sẽ tăng dần về số lượng và độ khó. Sau một khoản thời gian nhất định, sẽ có boss xuất hiện. Người chơi tiêu diệt boss sẽ nhận được nhiều kinh nghiệm hơn.



Hình 4.11 Cơ chế quái xuất hiện ngẫu nhiên

4.2.8 Cơ chế sinh tồn

Người chơi sẽ chơi liên tục cho đến khi nhân vật bị đánh bại hoặc tiêu diệt được boss cuối.



Hình 4.12 Khi nhân vật chết



Hình 4.13 Khi tiêu diệt được boss cuối

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Phát triển hoàn chỉnh một tựa game thể loại Roguelike với các nhân vật Genshin.

Có được những trải nghiệm, kinh nghiệm quý giá trong suốt quá trình làm game.

Cải thiện, nâng cao các kỹ năng lập trình, thiết kế đồ họa âm.

Rút ra được nhiều bài học cho các dự án sau này.

5.2. Hướng phát triển

Bổ sung thêm nhiều nhân vật Genshin khác, nâng cao chất lượng hình ảnh cho các nhân vật cũ.

Bổ xung thêm quái và boss.

Thêm map chơi mới.

Cải thiện lối chơi.

Cân bằng sức mạnh của các nhân vật và quái.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Wikipedia, "Roguelike - Wikipedia," [Online]. Available:
] <https://en.wikipedia.org/wiki/Roguelike>.
- [2] T. V. T. Huy, "Thiết kế game 2D bằng Unity," 2024.
- [3] GameMaker, "GML Code," [Online]. Available:
https://manual.gamemaker.io/monthly/en/GameMaker_Language/GameMaker_Language_Index.htm.
- [4] satellite1012, "Lập Trình Android Game Với Game Maker Studio," [Online].
Available: <https://codelearn.io/sharing/lap-trinh-game-maker-studio-p1>.
- [5] G. Code, "[Hướng dẫn GameMaker Platform] Wizard Story Eps.1," [Online].
Available: <https://www.youtube.com/watch?v=6rhg5JcCYHw>.
- [6] P. Burnham, "How to Make a Top Down Shooter in GameMaker Studio 2!,"
[Online]. Available:
https://www.youtube.com/watch?v=qTqDY4JtFfo&list=PL14Yj-e2sgzxTXIRYH-J2_PWAZRMahfLb.