

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA CÔNG NGHỆ PHẦN MỀM



ĐỒ ÁN MÔN HỌC
LẬP TRÌNH TRỰC QUAN
2D Pixel Shooter Game

- The Last Man -

Giảng viên hướng dẫn : Nguyễn Thị Xuân Hương
Nhóm : 4
Sinh viên thực hiện 1 : Ngô Đức Lộc - Leader
Mã sinh viên 1 : 22520790
Sinh viên thực hiện 2 : Bùi Duy Phúc
Mã sinh viên 2 : 22521108
Sinh viên thực hiện 3 : Đặng Thị Bảo Linh
Mã sinh viên 3 : 22520757
Lớp : IT008.O11
Bộ môn : Lập Trình Trực Quan

TP. HỒ CHÍ MINH, THÁNG 12 NĂM 2023

.....

BẢNG PHÂN CÔNG THỰC HIỆN ĐỒ ÁN MÔN HỌC		
Họ tên SV1: Ngô Đức Lộc MSSV: 22520790	Họ tên SV2: Bùi Duy Phúc MSSV: 22521108	Họ tên SV3: Đặng Thị Bảo Linh MSSV: 22520757
Thiết kế cơ sở dữ liệu	Thiết kế giao diện đầu game	Thiết kế giao diện cuối game
Thiết kế hệ thống game
Thiết kế đồ họa game
Thiết kế âm thanh		
SV thực hiện 1 (Ký tên)	SV thực hiện 2 (Ký tên)	SV thực hiện 3 (Ký tên)
Ngô Đức Lộc	Bùi Duy Phúc	Đặng Thị Bảo Linh

LỜI MỞ ĐẦU

Trên thế giới cũng như Việt Nam, công nghệ thông tin đã trở thành một công nghệ mũi nhọn, nó là ngành công nghệ kỹ thuật không thể thiếu trong việc áp dụng vào các hoạt động xã hội như: các hoạt động quản lý hệ thống, kinh doanh, trong hoạt động nghiên cứu,... và đặc biệt trong lĩnh vực Game - một lĩnh vực đang phát triển mạnh hiện nay.

Ở nước ta hiện nay, việc xây dựng một ứng dụng game đã không còn xa lạ, nhưng để tạo ra một game hay và chất lượng là một vấn đề không dễ. Đây là một vấn đề nan giải, để xây dựng được một tựa game đạt chất lượng ngoài việc người lập trình phải có kinh nghiệm, kiến thức vững chắc về các công cụ và ngôn ngữ lập trình, thì cần phải hiểu, hoạch định và phân tích tối ưu được các vai trò chức năng của hệ thống thông tin, chính vì thế mà nhóm em chọn đề tài “Xây dựng trò chơi 2D Pixel Shooter”

Game có thể loại là 2D Pixel Shooter và có tên là “The Last Man”, được tạo ra bởi trí sáng tạo và lấy cảm hứng về ngày hậu tận thế và mang tính giải trí đơn giản nên game là độc quyền, không có bản thứ hai trên thị trường.

Mặc dù đã rất cố gắng để hoàn thành công việc, nhưng do thiếu kinh nghiệm cũng như kỹ năng chưa cao nên việc phân tích và thiết kế còn nhiều thiếu sót, kính mong quý thầy cô và các bạn góp ý, bổ sung để nhóm hoàn thiện cho đồ án tốt hơn nữa. Nhóm chúng em xin chân thành cảm ơn!

Nhóm sinh viên thực hiện

Ngô Đức Lộc

Bùi Duy Phúc

Đặng Thị Bảo Linh

This image shows a full page of a document template designed for handwritten notes or essays. It features approximately 28 evenly spaced, thin grey horizontal lines extending across the entire width of the page. The margins are consistent on all sides, providing ample space for writing. There are no pre-printed questions, headings, or other markings on the page.

GVHD

MỤC LỤC

Chương 1. GIỚI THIỆU ĐỀ TÀI	11
1.1. Tên đề tài.....	11
1.2. Lý do chọn đề tài	11
1.3. Mô tả về game.....	11
1.4. Yêu cầu về sản phẩm	12
1.5. Các chức năng chính của đề tài	12
1.6. Công nghệ sử dụng.....	13
1.7. Môi trường lập trình	13
1.8. Công cụ hỗ trợ	13
Chương 2. GIỚI THIỆU CÔNG NGHỆ	14
2.1. Windows Form	14
2.1.1. Giới thiệu	14
2.1.2. Lịch sử.....	14
2.1.3. Chức năng.....	14
2.1.4. Áp dụng.....	14
2.2. SQL Server	15
2.2.1. Giới thiệu	15
2.2.2. Lịch sử.....	15
2.2.3. Chức năng.....	15
2.2.4. Áp dụng.....	15
Chương 3. THIẾT KẾ HỆ THỐNG	16

3.1. Khảo sát hiện trạng hệ thống	16
3.1.1. Xác định nguồn dữ liệu	16
3.1.2. Phân tích quá trình hoạt động	16
3.1.3. Đánh giá hiệu suất	16
3.1.4. Công cụ và công nghệ	16
3.1.5. Tạo thiết kế cơ sở dữ liệu	16
3.2. Mô hình hệ thống chức năng và xử lý	17
3.2.1. Các chức năng chính của hệ thống.....	17
3.2.1.1 Bắt đầu trò chơi.....	17
3.2.1.2 Tạo nhân vật và chọn map	17
3.2.1.3 Điều khiển	17
3.2.1.4 Tấn công.....	17
3.2.1.5 Phím tắt.....	17
3.2.1.6 Nâng cấp vũ khí.....	17
3.2.1.7 Mua vật phẩm.....	18
3.2.1.8 Hành trang.....	18
3.2.1.9 Tạm ngưng màn chơi.....	18
3.2.1.10 Kết thúc màn chơi.....	18
3.2.2. FlowCharts	19
3.2.2.1 Gameplay Flow Chart	19
3.2.2.2 Player Flow Chart.....	20
3.2.2.3 UpgradeWeapon Flow Chart.....	21

3.2.2.4	BuyItem Flow Chart	21
3.2.2.5	UseItem Flow Chart.....	22
3.3.	Thiết kế cơ sở dữ liệu	22
3.3.1.	Mô tả tân từ	22
3.3.2.	Mô hình thực thể kết hợp (ERD).....	23
3.3.3.	Database Diagram trong SQL	24
3.3.4.	Cấu trúc các bảng dữ liệu	25
3.3.5.	Dữ liệu mẫu	26
Chương 4.	XÂY DỰNG TRÒ CHƠI	28
4.1.	Mẫu thiết kế phần mềm (Design pattern).....	28
4.2.	Kết nối cơ sở dữ liệu (Database connection).....	29
4.3.	GameTimer	30
4.4.	Xử lý đồ họa (Graphic)	31
4.5.	Xử lý va chạm (Collision)	32
Chương 5.	GIAO DIỆN TRÒ CHƠI	34
5.1.	Màn hình bắt đầu	34
5.2.	Hướng dẫn	34
5.3.	Chế độ chơi – Cài đặt.....	36
5.4.	Kết nối cơ sở dữ liệu.....	36
5.5.	Chơi mới.....	37
5.6.	Chơi lại	37
5.7.	Màn hình chính	38

5.8. Cửa hàng	38
5.9. Nâng cấp vũ khí.....	39
5.10. Hành trang.....	39
5.11. Chiến thắng.....	40
5.12. Thất bại	40
5.13. Tạm ngưng.....	41
Chương 6. KẾT LUẬN	42
6.1. Kết quả	42
6.2. Đánh giá	43
6.2.1. Những điều đạt được	43
6.2.2. Những điều còn hạn chế	43
6.2.3. Hướng phát triển	43

DANH MỤC CÁC BẢNG BIỂU, HÌNH ẢNH

BẢNG

Bảng 3-1. Bảng dữ liệu.....	26
Bảng 3-2. Dữ liệu mẫu cho bảng Maps.....	26
Bảng 3-3. Dữ liệu mẫu cho bảng Weapons.....	27
Bảng 3-4. Dữ liệu mẫu cho bảng Items.....	27

HÌNH ẢNH

Hình 3-1. GamePlay Flow Chart.....	19
Hình 3-2. Player Flow Chart	20
Hình 3-3. UpgradeWeapon Flow Chart.....	21
Hình 3-4. BuyItem Flow Chart.....	21
Hình 3-5. UseItem Flow Chart	22
Hình 3-6. ER Diagram.....	24
Hình 3-7. Database Diagram	24
Hình 4-1. Singleton Pattern	28
Hình 4-2. Singleton Pattern Thread-Safe Code.....	29
Hình 4-3. SQL File	29
Hình 4-4. Connect Database and Get Data.....	30
Hình 4-5. GameTimer Code	31
Hình 4-6. Player.....	31
Hình 4-7. Mobs.....	31
Hình 4-8. GameCanvas Demo.....	32

Hình 4-9. GameObject Hitbox	32
Hình 4-10. Projectile Hitbox	32
Hình 4-11. Melee Attack	33
Hình 4-12. Shooting	33
Hình 5-1. Màn hình bắt đầu.....	34
Hình 5-2. Màn hình hướng dẫn chơi	35
Hình 5-3. Cài đặt chế độ chơi.....	36
Hình 5-4. Kết nối cơ sở dữ liệu	36
Hình 5-5. Màn hình chơi mới	37
Hình 5-6. Màn hình chơi lại	37
Hình 5-7. Màn hình chơi chính	38
Hình 5-8. Cửa hàng	38
Hình 5-9. Nâng cấp vũ khí	39
Hình 5-10. Hành trang	39
Hình 5-11. Màn chơi chiến thắng.....	40
Hình 5-12. Màn chơi thất bại.....	40
Hình 5-13. Tạm ngưng màn chơi	41
Hình 6-1. Windows Forms	42

Chương 1. GIỚI THIỆU ĐỀ TÀI

1.1. Tên đề tài

Game có tên là “**The Last Man**”, được tạo ra bởi trí sáng tạo và lấy cảm hứng về thể loại game bắn súng, yếu tố kinh dị từ ngày hậu tận thế và mang tính giải trí đơn giản với phong cách 2D pixel art.

1.2. Lý do chọn đề tài

Trên thế giới cũng như Việt Nam, công nghệ thông tin đã trở thành một công nghệ mũi nhọn, nó là ngành công nghệ kỹ thuật không thể thiếu trong việc áp dụng vào các hoạt động xã hội như: các hoạt động quản lý hệ thống, kinh doanh, trong hoạt động nghiên cứu... và đặc biệt trong lĩnh vực Game - Một lĩnh vực đang phát triển mạnh hiện nay.

Ở nước ta hiện nay, việc xây dựng một ứng dụng game đã không còn xa lạ, nhưng để tạo ra một game hay và chất lượng là một vấn đề không dễ. Đây là một vấn đề nan giải, để xây dựng được một tựa game đạt chất lượng ngoài việc người lập trình phải có kinh nghiệm, kiến thức vững chắc về các công cụ và ngôn ngữ lập trình, thì cần phải hiểu, hoạch định và phân tích tối ưu được các vai trò chức năng của hệ thống thông tin, chính vì thế mà nhóm em chọn đề tài “Phân tích và thiết kế hệ thống game”

1.3. Mô tả về game

Đến với The Last Man, người chơi sẽ nhập vai vào nhân vật với nhiệm vụ là sống sót sau 10 vòng chơi bị quái vật truy đuổi, sau khi bấm vào ô bắt đầu chơi thì người chơi sẽ được đưa đến map đã chọn với vòng đầu tiên là 1. Người chơi có thể sử dụng vũ khí cận chiến hoặc vũ khí tầm xa. Người chơi sẽ có thể hồi máu hoặc mua đạn thông qua cửa hàng (Shop) hoặc nâng cấp vũ khí bằng cửa sổ nâng cấp vũ khí (Upgrade Weapon), với đơn vị tiền tệ là gold.

Mỗi vòng hệ thống sẽ sản sinh quái vật dạng thường (Normal Mob) để truy đuổi tấn công người chơi. Sau khi tiêu diệt toàn bộ thì sẽ tăng lên một vòng và độ khó cũng tăng

theo. Tại vòng cuối thì sẽ có thêm một quái vật siêu cấp (Elite Mob) có chỉ số vượt bậc, khả năng tấn công tầm xa, và đặc biệt là biết giữ khoảng cách với người chơi. Sau khi tiêu diệt một quái vật thì số vàng của người chơi sẽ tăng ngẫu nhiên trong khoảng dựa trên cấp bậc của quái vật.

Sau khi người vượt thành công 10 vòng, hệ thống sẽ hiển thị lời chúc mừng người chơi đã hoàn thành trò chơi ở bản đồ đã chọn, thời gian tạo nhân vật, thời gian hoàn thành màn chơi, và sau đó tự động lưu vào lịch sử chơi là màn chơi đó đã chiến thắng. Nếu người chơi thất bại, thì hệ thống sẽ lưu lại màn chơi mà người chơi đó đã thất bại vào hệ thống, người chơi có thể chơi lại màn chơi đó sau này.

1.4. Yêu cầu về sản phẩm

- Game phải có dung lượng không quá lớn.
- Tốc độ xử lý nhanh.
- Giao diện game dễ nhìn, thân thiện với người sử dụng.
- Công việc tính toán (xử lý va chạm, điểm,) phải thực hiện chính xác.
- Tạo cảm giác sống động cho người chơi.
- Phân tích game theo hướng đối tượng cụ thể, rõ ràng.

1.5. Các chức năng chính của đề tài

- Cho phép người dùng tạo mới màn chơi và xem lịch sử đã chơi
- Cho phép người chơi chọn độ khó của màn chơi
- Người chơi có thể chọn bản đồ (map) cho màn chơi
- Cho phép người dùng có thể điều khiển nhân vật thông qua các phím di chuyển và tấn công bằng các phím tắt
- Có thể ngưng màn chơi tạm thời bằng nút Esc
- Hiển thị chuyển động với mức độ 60 fps

1.6. Công nghệ sử dụng

- Ngôn ngữ lập trình C# kết hợp với CSDL của SQL Server
- Lập trình Windows Winform.

1.7. Môi trường lập trình

- Microsoft Visual Studio 2022

1.8. Công cụ hỗ trợ

- SQL Server Management Studio: Xây dựng CSDL
- Github: Quản lý project

Chương 2. GIỚI THIỆU CÔNG NGHỆ

2.1. Windows Form

2.1.1. Giới thiệu

Windows Forms (WinForms) là GUI mã nguồn mở và miễn phí được bao gồm như một phần của Microsoft.NET Framework hoặc Mono Framework, cung cấp nền tảng để viết các ứng dụng khách phong phú cho máy tính. Mặc dù nó được coi là sự thay thế cho Thư viện lớp nền tảng Microsoft Foundation của C++ trước đây và phức tạp hơn, nhưng nó không cung cấp mô hình tương đương và chỉ hoạt động như một nền tảng cho tầng giao diện người dùng trong một giải pháp nhiều tầng.

2.1.2. Lịch sử

Windows Forms phát hành lần đầu vào 13 - 2 - 2002. Giống như Tóm tắt Window Toolkit (AWT), API Java tương đương, các biểu mẫu Windows là một cách sớm và dễ dàng để cung cấp các thành phần giao diện người dùng đồ họa cho .NET Framework. Các biểu mẫu Windows được xây dựng trên API Windows hiện có và một số điều khiển chỉ quản các thành phần Windows bên dưới. Một số phương thức cho phép truy cập trực tiếp vào các cuộc gọi lại Win32, không khả dụng trong các nền tảng không phải Windows.

2.1.3. Chức năng

Cung cấp các control hỗ trợ tính năng kéo thả để thiết kế giao diện người dùng, nhờ vậy, việc xây dựng, phát triển các ứng dụng trở nên dễ dàng.

2.1.4. Áp dụng

Thiết kế cũng như xây dựng các form và screen trong trò chơi như là: GameWindow, SettingsScreen, PlayNewScreen, LoadPlayScreen,...

2.2. SQL Server

2.2.1. Giới thiệu

Microsoft SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ được phát triển bởi Microsoft. Là một máy chủ cơ sở dữ liệu, nó là một sản phẩm phần mềm có chức năng chính là lưu trữ và truy xuất dữ liệu theo yêu cầu của các ứng dụng phần mềm khác. Có thể chạy trên cùng một máy tính hoặc trên một máy tính khác trên mạng (bao gồm cả Internet).

2.2.2. Lịch sử

Microsoft SQL Server bắt đầu với sản phẩm Microsoft SQL Server đầu tiên SQL Server 1.0, máy chủ 16-bit cho hệ điều hành OS/2 vào năm 1989 và kéo dài đến ngày hiện tại.

2.2.3. Chức năng

Lưu trữ và truy xuất dữ liệu theo yêu cầu của các ứng dụng phần mềm khác.

2.2.4. Áp dụng

Tạo cơ sở dữ liệu GAME dùng để lưu trữ phần lớn các thông tin của ứng dụng ví dụ như: Thông tin người chơi, thông tin các vũ khí, bản đồ, vật phẩm, thông tin các màn chơi của người chơi,...

Chương 3. THIẾT KẾ HỆ THỐNG

3.1. Khảo sát hiện trạng hệ thống

3.1.1. Xác định nguồn dữ liệu

- Dữ liệu sẽ được lưu vào máy của người chơi.

3.1.2. Phân tích quá trình hoạt động

- Quá trình tạo mới nhân vật
- Quá trình lựa chọn bản đồ (map) và độ khó (difficulty)
- Quá trình nâng cấp vũ khí
- Quá trình mua vật phẩm hỗ trợ

3.1.3. Đánh giá hiệu suất

- Đảm bảo rằng cơ sở dữ liệu có khả năng xử lý số lượng lớn các sự kiện game đồng thời

3.1.4. Công cụ và công nghệ

- Cơ sở dữ liệu SQL.
- Ngôn ngữ lập trình C# và IDE Microsoft Visual Studio để kết nối cơ sở dữ liệu và lưu trữ dữ liệu

3.1.5. Tạo thiết kế cơ sở dữ liệu

- Xây dựng biểu đồ ERD để mô tả cấu trúc dữ liệu.
- Xây dựng mô tả chi tiết của các bảng và các quan hệ của chúng.

3.2. Mô hình hệ thống chức năng và xử lý

3.2.1. Các chức năng chính của hệ thống

3.2.1.1 Bắt đầu trò chơi

Giải thích: Để bắt đầu trò chơi, bấm vào nút PlayNew nếu bạn muốn chơi mới hoặc LoadPlay để chơi lại màn chơi đã thất bại.

3.2.1.2 Tạo nhân vật và chọn map

Giải thích: Khi chơi mới, người chơi điền tên cho nhân vật và chọn bản đồ (map) để chơi. Hệ thống sẽ lưu thông tin vào cơ sở dữ liệu.

3.2.1.3 Điều khiển

Giải thích: Người chơi điều khiển nhân vật bằng bốn phím di chuyển: ←, ↑, →, ↓.

3.2.1.4 Tấn công

Giải thích: Người chơi thực hiện tấn công thông qua phím “A”.

3.2.1.5 Phím tắt

Giải thích: Hệ thống trò chơi có các phím tắt để cho người chơi thao tác dễ dàng hơn:

- S - Mở cửa hàng vật phẩm.
- I - Mở hành trang
- U - Mở mục nâng cấp vũ khí hiện tại.
- ESC – Tạm ngưng màn chơi

3.2.1.6 Nâng cấp vũ khí

Giải thích: Khi người chơi nhấn phím U, cửa sổ nâng cấp vũ khí sẽ hiện ra. Người chơi có thể nâng cấp vũ khí hiện tại của bản thân, hệ thống sẽ xác nhận thao tác có thành công hay không dựa trên số vàng của người chơi và giá nâng cấp. Nếu thành công, hệ thống sẽ cập nhật mã vũ khí cho người chơi theo thời gian thực vào cơ sở dữ liệu.

3.2.1.7 Mua vật phẩm

Giải thích: Khi người chơi nhấn phím S, cửa sổ cửa hàng vật phẩm sẽ hiện ra. Người chơi có thể mua vật phẩm hỗ trợ, hệ thống sẽ xác nhận thao tác có thành công hay không dựa trên số vàng của người chơi và giá của vật phẩm. Nếu thành công, hệ thống sẽ cập nhật thông tin vật phẩm đã được người chơi mua vào cơ sở dữ liệu.

3.2.1.8 Hành trang

Giải thích: Khi người chơi nhấn phím I, cửa sổ hành trang sẽ hiện ra. Người chơi có thể dùng các vật phẩm của mình đã mua, hệ thống sẽ xác nhận thao tác có thành công hay không dựa trên số lượng còn lại của vật phẩm. Nếu thành công, hệ thống sẽ cập nhật số lượng vật phẩm còn lại vào cơ sở dữ liệu.

3.2.1.9 Tạm ngưng màn chơi

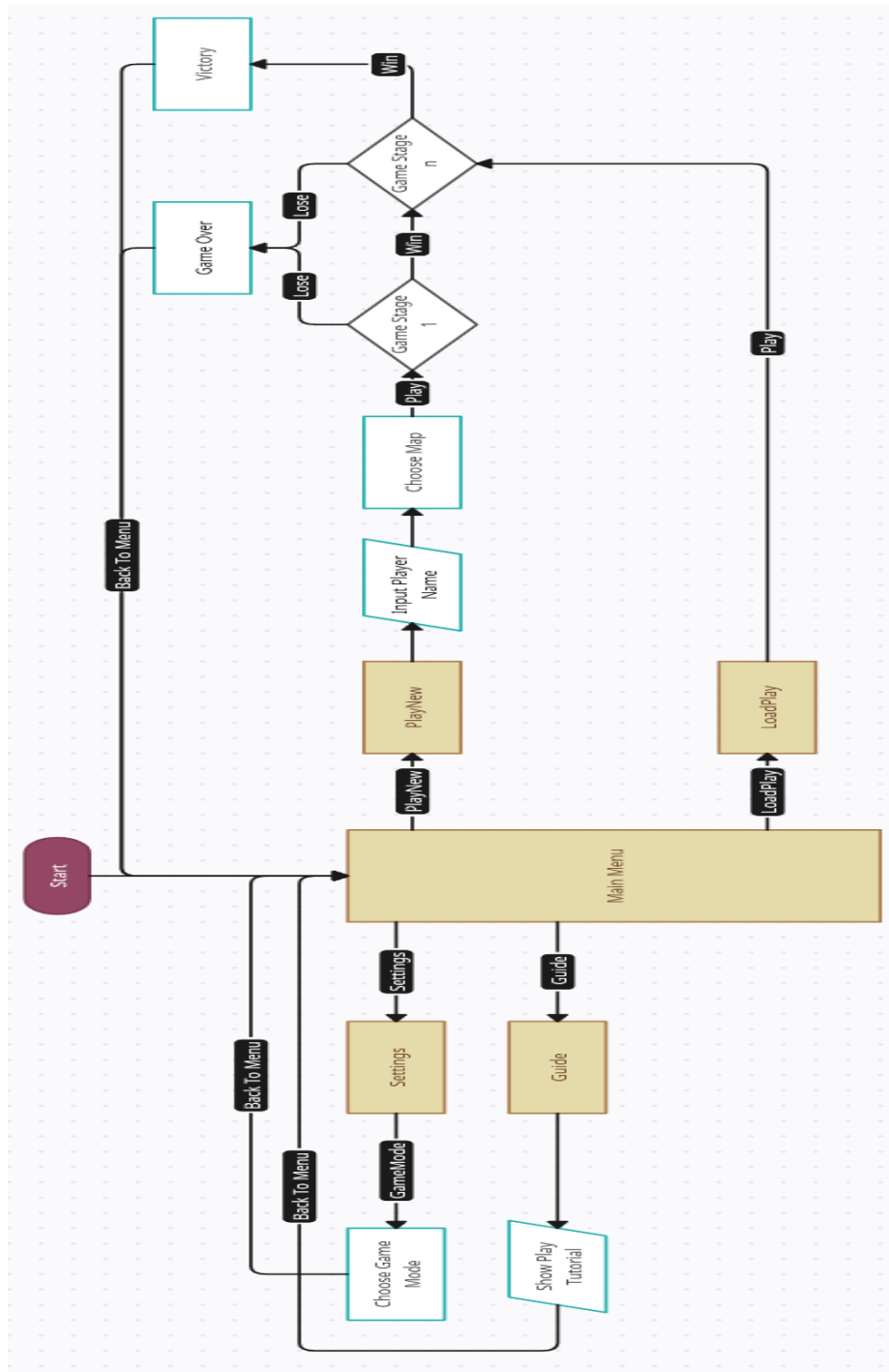
Giải thích: Khi người chơi nhấn phím ESC, màn chơi sẽ được tạm ngưng và sẽ xuất hiện một cửa sổ với hai nút là “Resume” – dùng để tiếp tục màn chơi hoặc “Back to menu” – dùng để thoát màn chơi và quay về màn hình menu chính.

3.2.1.10 Kết thúc màn chơi

Giải thích: Khi người chơi bị tấn công và giảm máu (HP) đến mức 0 hoặc thấp hơn thì sẽ được xác nhận là thất bại, hoặc người chơi tiêu diệt thành công toàn bộ quái vật trong 10 vòng chơi thì sẽ được xác nhận là chiến thắng. Hệ thống sẽ lưu lại màn chơi của người chơi ở cơ sở dữ liệu. Số lần chơi lại sẽ tăng lên mỗi khi người chơi chọn chơi lại màn chơi đó và dữ liệu sẽ được hệ thống cập nhật vào cơ sở dữ liệu.

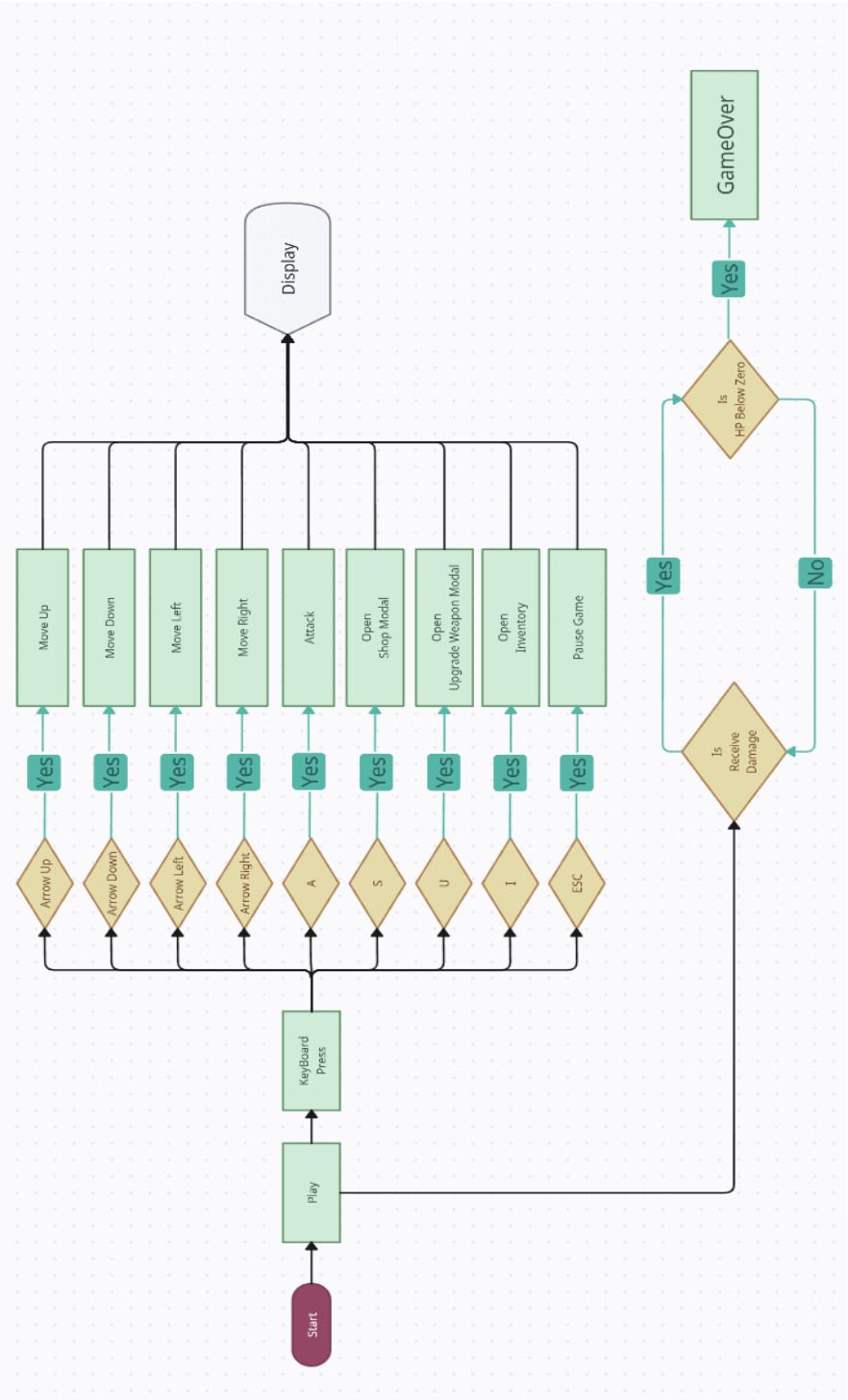
3.2.2. FlowCharts

3.2.2.1 GamePlay Flow Chart



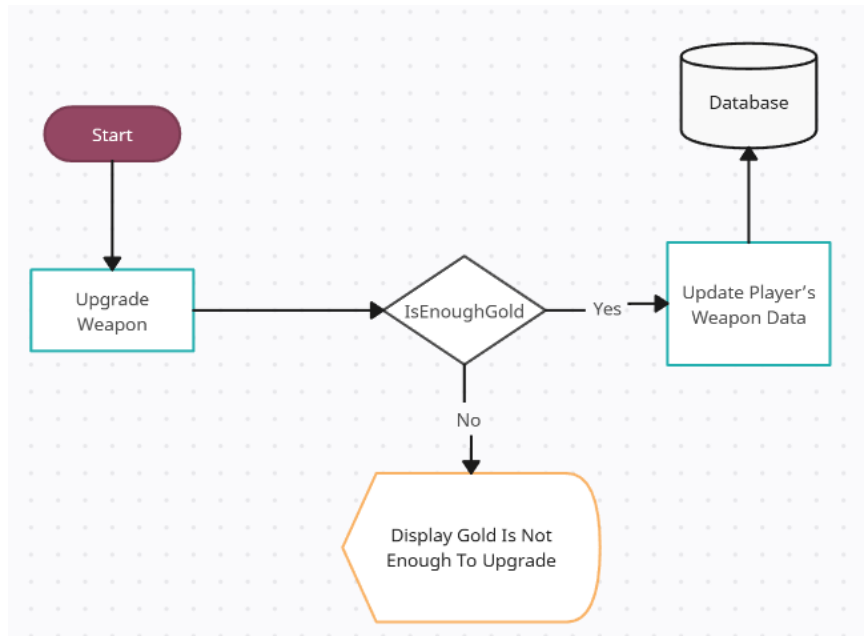
Hình 3-1. GamePlay Flow Chart

3.2.2.2 Player Flow Chart



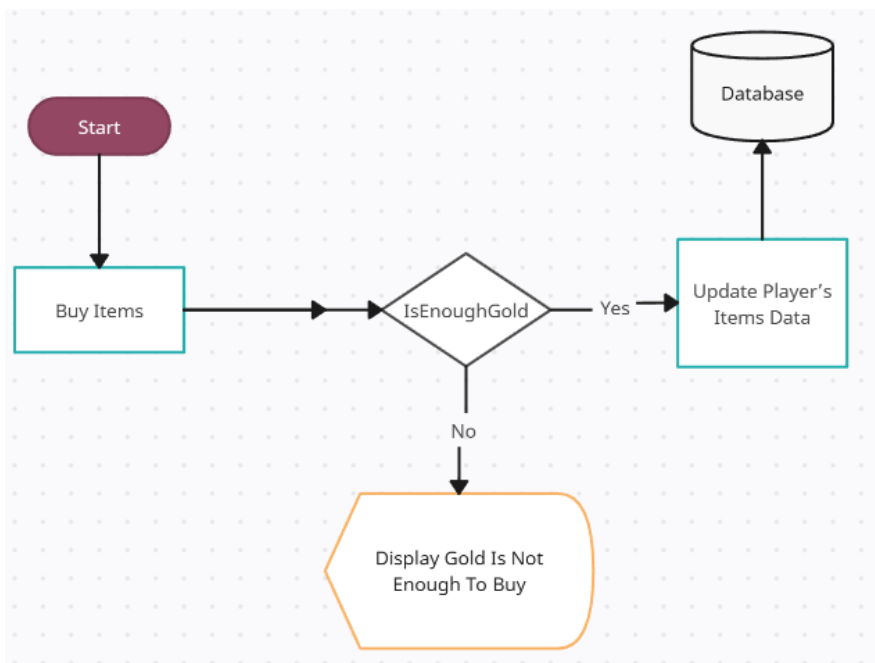
Hình 3-2. Player Flow Chart

3.2.2.3 UpgradeWeapon Flow Chart



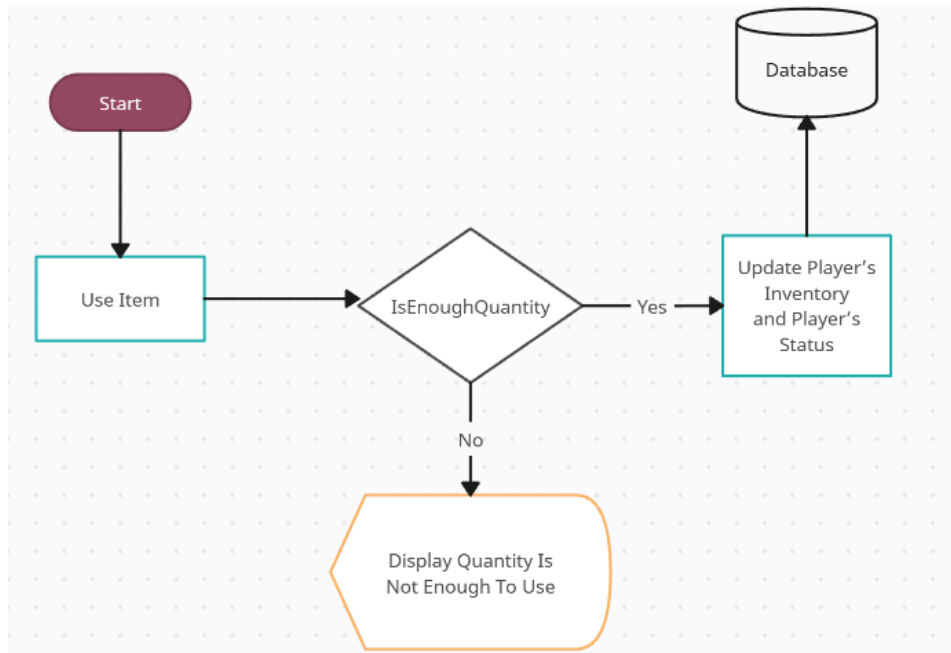
Hình 3-3. UpgradeWeapon Flow Chart

3.2.2.4 BuyItem Flow Chart



Hình 3-4. BuyItem Flow Chart

3.2.2.5 UseItem Flow Chart



Hình 3-5. UseItem Flow Chart

3.3. Thiết kế cơ sở dữ liệu

3.3.1. Mô tả tân từ

- Mỗi nhân vật (Players) sẽ có một mã nhân vật (Player_ID) duy nhất để phân biệt, mỗi nhân vật sẽ có tên (Player_Name), lượng máu (HP), số đạn (Ammo), số vàng (Gold), và thời điểm khởi tạo nhân vật (Created_At).
- Mỗi bản đồ (Maps) sẽ có mã bản đồ (Map_ID) duy nhất để phân biệt, tên bản đồ (Map_Name)
- Mỗi vũ khí (Weapons) sẽ có mã vũ khí (Weapon_ID) duy nhất để phân biệt, tên vũ khí (Weapon_Name), lượng sát thương vũ khí gây ra (Damage), giá mua vũ khí (Price)
- Mỗi vật phẩm (Items) sẽ có mã vật phẩm (Item_ID) duy nhất để phân biệt, tên vật phẩm (Item_Name), thông tin của vật phẩm (Item_Info), giá mua vật phẩm (Price)

- Màn chơi (Play) sẽ có mã người chơi (Player_ID) và mã bản đồ (Map_ID), Độ khó (Difficulty), Số vòng hiện tại (Stage), Kết quả màn chơi (isVictory), Số lần chơi (Attempt). Mỗi nhân vật có thể tạo nhiều màn chơi ở nhiều bản đồ, một bản đồ có thể được chơi bởi nhiều nhân vật.
- Sở hữu (Own) sẽ có mã người chơi (Player_ID) và mã vũ khí (Weapon_ID). Mỗi người chơi có thể mua và sở hữu nhiều vũ khí. Mỗi vũ khí có thể được sở hữu bởi nhiều nhân vật.
- Hành trang (Inventory) sẽ có mã người người (Player_ID), mã vật phẩm (Item_ID), và số lượng (Quantity). Mỗi người chơi có thể mua và sở hữu nhiều vật phẩm với số lượng. Mỗi vật phẩm có thể được mua bởi nhiều nhân vật.

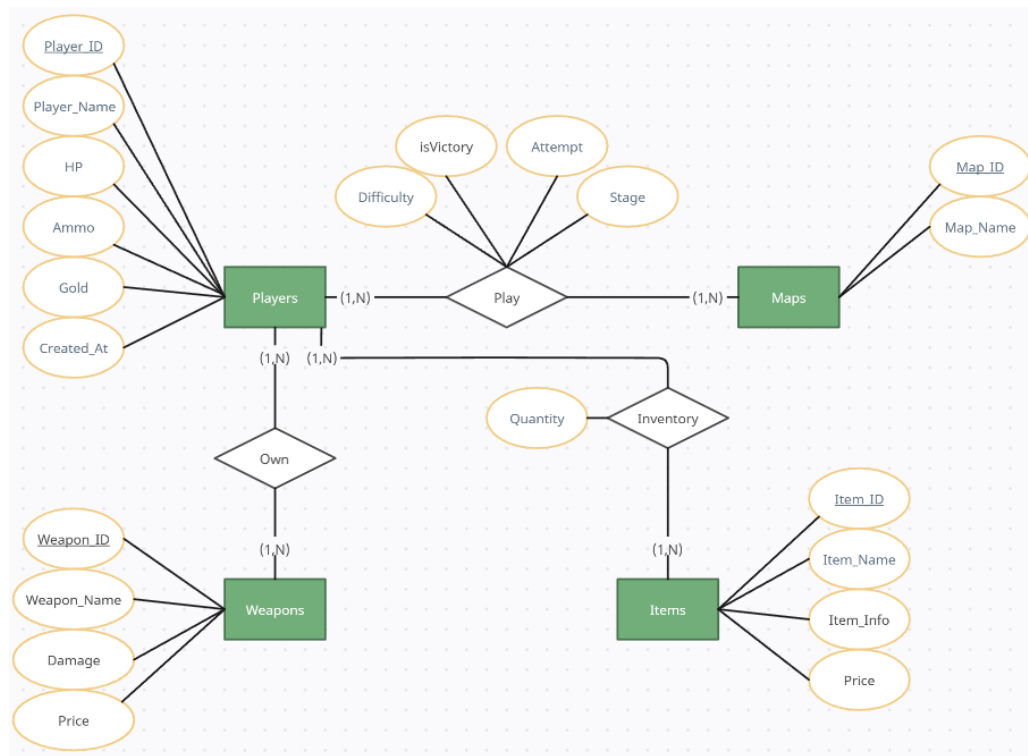
3.3.2. Mô hình thực thể kết hợp (ERD)

Có bốn thực thể chính:

- Players: Player_ID (Khóa chính), Player_Name, HP, Ammo, Gold, Created_At.
- Maps: Map_ID (Khóa chính), Map_Name.
- Weapons: Weapon_ID (Khóa chính), Weapon_Name, Damage, Price.
- Items: Item_ID (Khóa chính), Item_Name, Item_Info, Price.

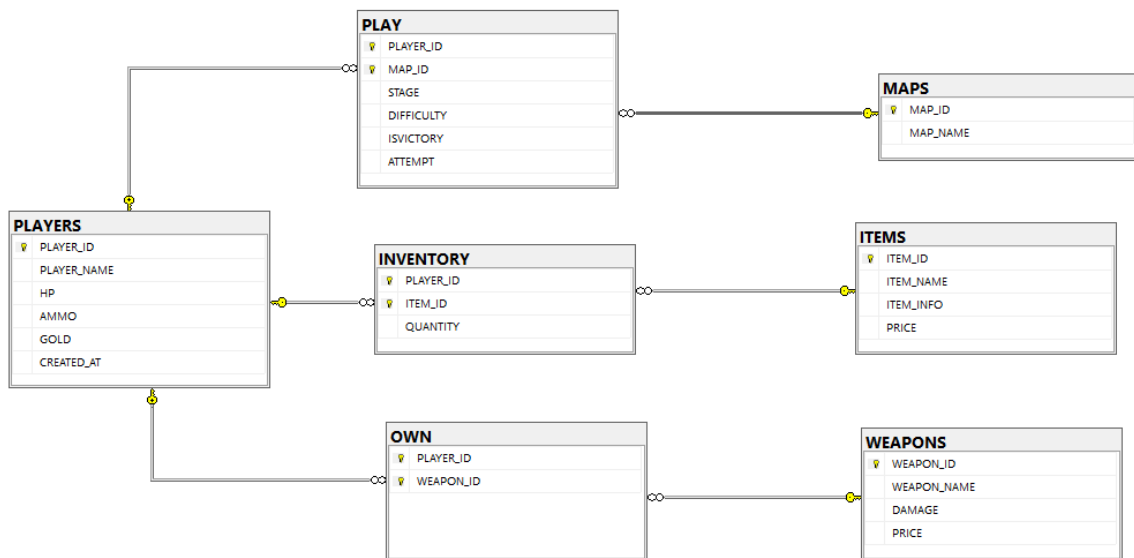
Có ba mối kết hợp:

- Play: Mối kết hợp giữa Players và Maps, có thuộc tính isVictory, Stage, Difficulty, Attempt.
- Own: Mối kết hợp giữa Players và Weapons.
- Inventory: Mối kết hợp giữa Players và Items, có thuộc tính Quantity



Hình 3-6. ER Diagram

3.3.3. Database Diagram trong SQL



Hình 3-7. Database Diagram

3.3.4. Cấu trúc các bảng dữ liệu

Quan hệ	Thuộc tính	Diễn giải	Kiểu dữ liệu
Players	Player_ID	Mã người chơi	Varchar(3)
	Player_Name	Tên người chơi	Varchar(20)
	HP	Lượng máu của người chơi	Int
	Ammo	Số lượng đạn của người chơi	Int
	Gold	Số lượng vàng của người chơi	Int
	Created_At	Thời điểm người chơi tạo nhân vật	Smalldatetime
Maps	Map_ID	Mã bản đồ	Varchar(3)
	Map_Name	Tên bản đồ	Varchar(20)
Items	Item_ID	Mã vật phẩm	Varchar(3)
	Item_Name	Tên vật phẩm	Varchar(20)
	Item_Info	Thông tin tác dụng của vật phẩm	Varchar(100)
	Price	Giá của vật phẩm	Int
Weapons	Weapon_ID	Mã vũ khí	Varchar(3)
	Weapon_Name	Tên vũ khí	Varchar(20)
	Damage	Lượng sát thương của vũ khí	Int
	Price	Giá của vũ khí	Int
Play	Player_ID	Mã người chơi	Varchar(3)
	Map_ID	Mã bản đồ	Varchar(3)
	Stage	Vòng chơi hiện tại	Int

	Difficulty	Độ khó của màn chơi	Int
	isVictory	Kết quả của màn chơi	Char(1)
	Attempt	Số lần chơi màn chơi	Int
Inventory	Player_ID	Mã người chơi	Varchar(3)
	Item_ID	Mã vật phẩm	Varchar(3)
	Quantity	Số lượng vật phẩm	Int
Own	Player_ID	Mã người chơi	Varchar(3)
	Weapon_ID	Mã vũ khí	Varchar(3)

Bảng 3-1. Bảng dữ liệu

3.3.5. Dữ liệu mẫu

MAPS	
Map_ID	Map_Name
M01	Ruin Street

Bảng 3-2. Dữ liệu mẫu cho bảng Maps

WEAPONS			
Weapon_ID	Weapon_Name	Damage	Price
W01	Knife	1	0
W02	Pistol	2	2
W03	AK-47	5	5
W04	RPG	10	10

W05	Bazooka	50	50
-----	---------	----	----

Bảng 3-3. Dữ liệu mẫu cho bảng Weapons

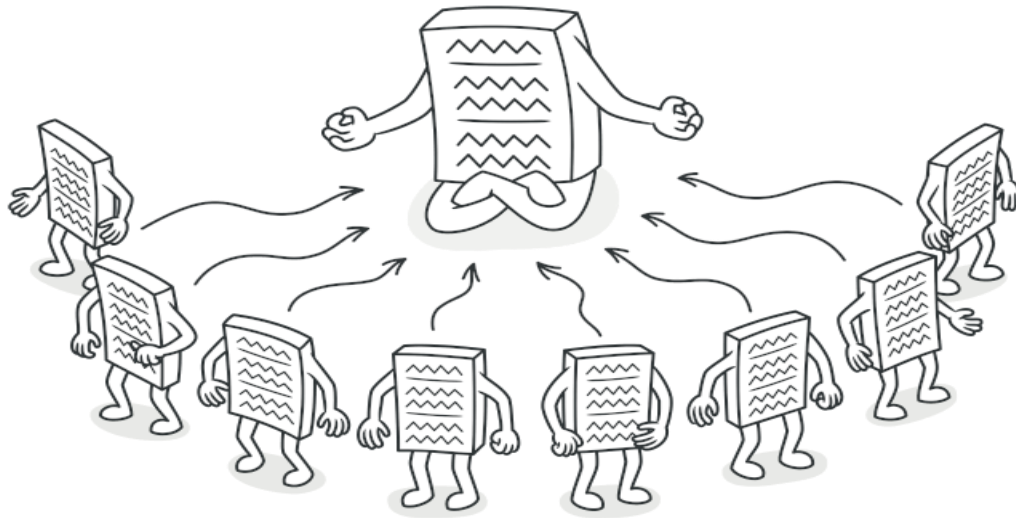
ITEMS			
Item_ID	Item_Name	Item_Info	Price
I01	Ammo Pack 1	+5 Ammo	1
I02	Ammo Pack 2	+20 Ammo	3
I03	Speed Potion	+1 Speed	10
I04	Health Potion	+10 HP	20

Bảng 3-4. Dữ liệu mẫu cho bảng Items

Chương 4. XÂY DỰNG TRÒ CHƠI

4.1. Mẫu thiết kế phần mềm (Design pattern)

Trò chơi sử dụng Singleton Pattern cho các lớp màn hình game (GameWindows), điều khiển cơ sở dữ liệu (Database Controller) và tải dữ liệu ảnh và âm thanh (AssetsLoader) vì các lớp này chỉ cần duy nhất một instance để thực thi công việc trong suốt quá trình hoạt động của chương trình. Nhờ có ưu điểm của Singleton Pattern mà có thể dễ dàng cài đặt và sử dụng dữ liệu một cách tối ưu.



Hình 4-1. Singleton Pattern

Các lớp sử dụng Singleton Pattern đều được viết theo cấu trúc thread-safe để có thể tối ưu được tính đa luồng của hệ thống:

```

public class DatabaseController
{
    1 reference
    private DatabaseController() { }
    private static DatabaseController instance;
    private static readonly object _lock = new object();
    public SqlConnection sqlConnection;
    private SqlDataAdapter sqlDataAdapter;
    public DataSet DataSet;
    50 references
    public static DatabaseController Instance
    {
        get
        {
            if(instance == null)
            {
                lock(_lock)
                {
                    if(instance==null)
                    {
                        instance = new DatabaseController();
                        instance.DataSet = new DataSet();
                    }
                }
            }
            return instance;
        }
        private set { instance = value; }
    }
    11 references
    public void GetAllTablesData()...
    1 reference
    public void DoConnect(string Server, string Database, string Trusted, string UID, string Pwd)...)
    //Players
    1 reference
    public string InsertPlayerData(string Pname)...)
    1 reference
    public List<string> GetPlayerData(string PID)...)
    1 reference
    public string GetPlayerName(string PID)...)
    1 reference
    public string GetPlayerCreateTime(string PID)...)
    6 references
    public void UpdatePlayerHP(string PID, int HP)...)

```

Hình 4-2. Singleton Pattern Thread-Safe Code

4.2. Kết nối cơ sở dữ liệu (Database connection)

- SQL file và dữ liệu mẫu:

```

|CREATE DATABASE GAME
|USE GAME
|CREATE TABLE PLAYERS(...)

|CREATE TABLE MAPS...
|INSERT INTO MAPS VALUES ('M01','Ruin Street')

|CREATE TABLE PLAY...

|CREATE TABLE WEAPONS(...)
|INSERT INTO WEAPONS VALUES('W01','Knife',1,0);
|INSERT INTO WEAPONS VALUES('W02','Pistol',2,2);
|INSERT INTO WEAPONS VALUES('W03','AK-47',5,5);
|INSERT INTO WEAPONS VALUES('W04','RPG',10,10);
|INSERT INTO WEAPONS VALUES('W05','Bazooka',50,50);

|CREATE TABLE ITEMS(...)
|INSERT INTO ITEMS VALUES ('I01','Ammo Pack 1','+5 Ammo',1)
|INSERT INTO ITEMS VALUES ('I02','Ammo Pack 2','+20 Ammo',3)
|INSERT INTO ITEMS VALUES ('I03','Speed Potion','+1 Speed',10)
|INSERT INTO ITEMS VALUES ('I04','Health Potion','+10 HP',20)

|CREATE TABLE INVENTORY...

|CREATE TABLE OWN...

```

Hình 4-3. SQL File

- Thao tác kết nối cơ sở dữ liệu và lấy dữ liệu của toàn bộ bảng:

```

public void GetAllTablesData()
{
    sqlDataAdapter = new SqlDataAdapter();
    string cmd = "SELECT * FROM PLAYERS;" +
        "SELECT * FROM MAPS;" +
        "SELECT * FROM WEAPONS;" +
        "SELECT * FROM ITEMS;" +
        "SELECT PLAYER_NAME, CREATED_AT, PLAY.* FROM PLAY, PLAYERS WHERE PLAY.PLAYER_ID = PLAYERS.PLAYER_ID;" +
        "SELECT * FROM OWN;" +
        "SELECT * FROM INVENTORY;";
    sqlDataAdapter.MissingSchemaAction = MissingSchemaAction.AddWithKey;
    try
    {
        sqlDataAdapter.SelectCommand = new SqlCommand(cmd, sqlConnection);
        sqlDataAdapter.TableMappings.Add("Table", "PLAYERS");
        sqlDataAdapter.TableMappings.Add("Table1", "MAPS");
        sqlDataAdapter.TableMappings.Add("Table2", "WEAPONS");
        sqlDataAdapter.TableMappings.Add("Table3", "ITEMS");
        sqlDataAdapter.TableMappings.Add("Table4", "PLAY");
        sqlDataAdapter.TableMappings.Add("Table5", "OWN");
        sqlDataAdapter.TableMappings.Add("Table6", "INVENTORY");
        sqlDataAdapter.Fill(this.DataSet);
    }
    catch (Exception e)
    {
        MessageBox.Show(e.ToString());
    }
}

1 reference
public void DoConnect(string Server, string Database, string Trusted, string UID, string PWD)
{
    var strbuilder = new SqlConnectionStringBuilder();
    strbuilder["Server"] = Server;
    strbuilder["Database"] = Database;
    strbuilder["Trusted_Connection"] = Trusted;
    if (Trusted == "false")
    {
        strbuilder["UID"] = UID;
        strbuilder["PWD"] = PWD;
    }
    this.sqlConnection = new SqlConnection(strbuilder.ToString());
    if (this.sqlConnection.State != ConnectionState.Open)
    {
        DoOpenConnection();
        GetAllTablesData();
    }
}

```

Hình 4-4. Connect Database and Get Data

4.3. GameTimer

Trong Visual Studio sẽ có hai loại Timer:

System.Timers.Timers	System.Windows.Forms.Timer
Thiết lập đối tượng đồng bộ hóa cho một điều khiển UI khiến sự kiện được kích hoạt trên luồng UI của điều khiển đó.	Chỉ hoạt động trên luồng UI
Được thiết kế cho các ứng dụng đa luồng và do đó an toàn đối với luồng thông qua thuộc tính SynchronizationObject	Không được thiết kế đa luồng và không thread-safe

⇒ Do trò chơi được xây dựng theo hướng đa luồng nên việc sử dụng **System.Timers.Timer** là rất cần thiết khi muốn hệ thống xử lý đồ họa và các vấn đề khác với tốc độ nhanh.

```
using GameTimer = System.Timers.Timer;

public GameTimer gameTimer = new GameTimer();

private void SetGameTimer()
{
    gameTimer.Enabled = true;
    gameTimer.AutoReset = true;
    gameTimer.Interval = 16;
    gameTimer.SynchronizingObject = gameCanvas;
}
```

Hình 4-5. GameTimer Code

4.4. Xử lý đồ họa (Graphic)

- Hệ thống sử dụng các chức năng như Graphic để vẽ nên các hình ảnh, và sử dụng ImageAnimator để có thể hiển thị các hoạt ảnh gif

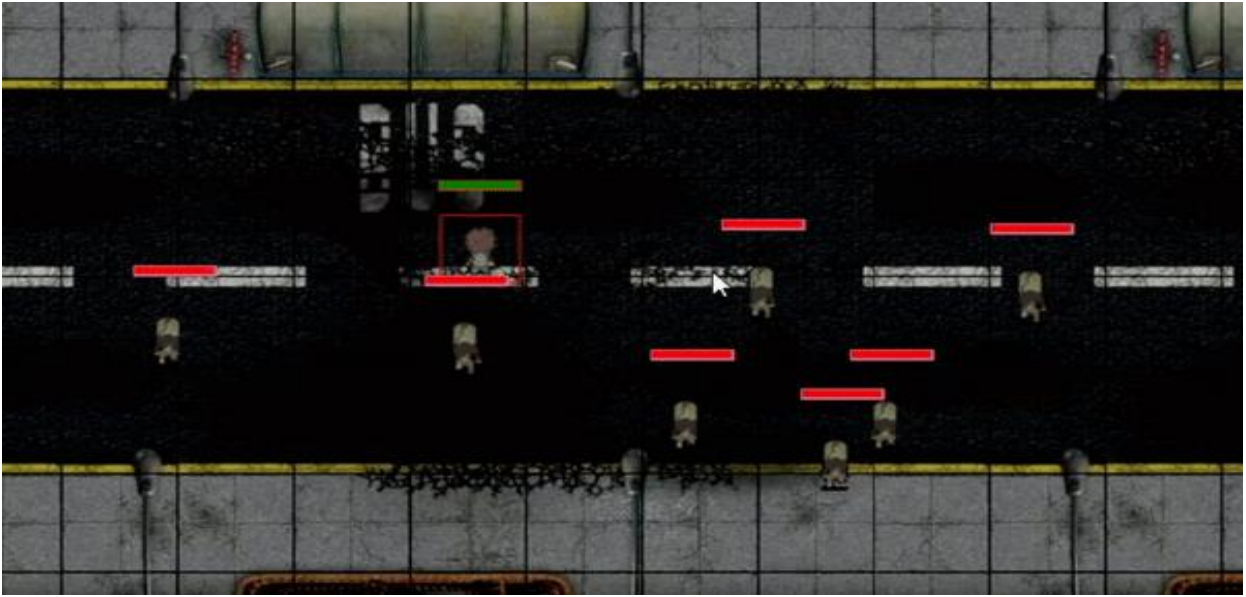


Hình 4-6. Player



Hình 4-7. Mobs

- Nhờ có ImageAnimator và GameTimer, khung hình chơi (GameCanvas) sẽ được cập nhật liên tục xấp xỉ 60 frame 1 giây (60FPS) để hoạt ảnh có thể hiển thị và di chuyển trong trò chơi mượt mà hơn



Hình 4-8. GameCanvas Demo

4.5. Xử lý va chạm (Collision)

- Để có thể xử lý va chạm thì mỗi đối tượng trong game đều phải có riêng hitbox để có thể xác nhận va chạm. Hệ thống sẽ tạo một hình chữ nhật Hitbox cho mỗi đối tượng và cập nhật vị trí theo vị trí của đối tượng đó theo thời gian
- GameObject HitBox:



Hình 4-9. GameObject Hitbox

- Projectile HitBox:



Hình 4-10. Projectile Hitbox

- Việc xử lý va chạm dựa trên việc giao nhau giữa các HitBox:

- Nếu vật thể bay (projectile) va chạm vào HitBox của đối tượng game thì hệ thống sẽ tính toán sát thương và tiêu hủy đi hình ảnh của vật thể bay lẫn hitbox của nó.
- Nếu người chơi dùng vũ khí cận chiến thì chỉ khi giao nhau với hitbox của quái vật thì mới gây ra sát thương dựa trên phím tấn công và quái vật cũng tương tự thế.



Hình 4-11. Melee Attack



Hình 4-12. Shooting

Chương 5. GIAO DIỆN TRÒ CHƠI

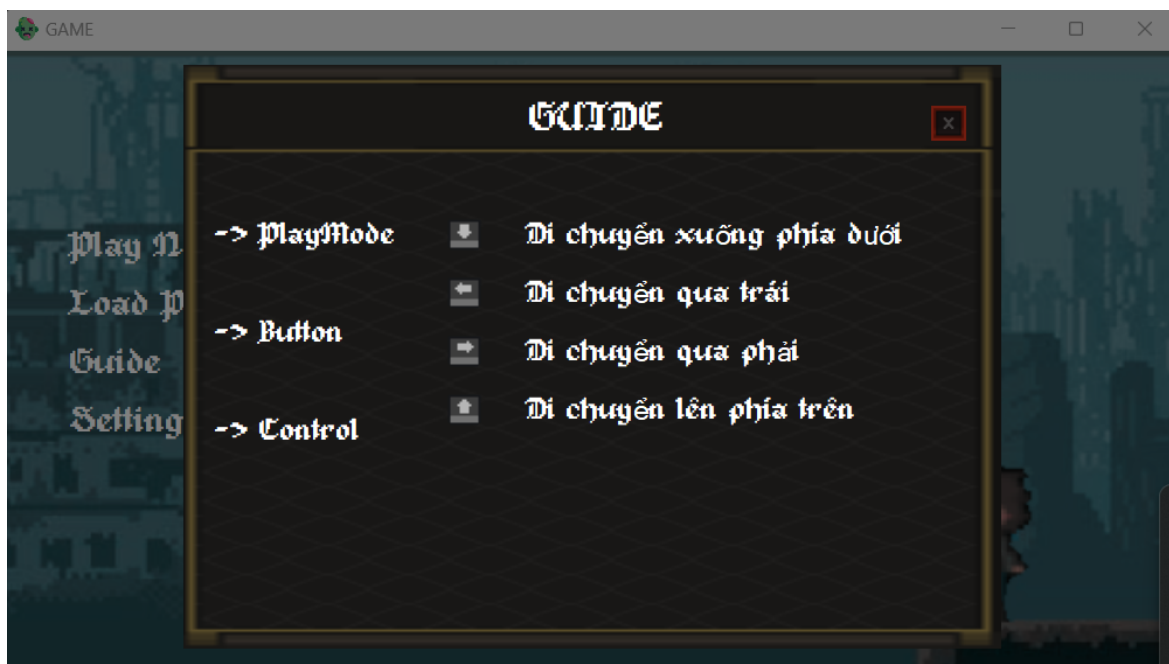
5.1. Màn hình bắt đầu



Hình 5-1. Màn hình bắt đầu

5.2. Hướng dẫn





Hình 5-2. Màn hình hướng dẫn chơi

5.3. Chế độ chơi – Cài đặt



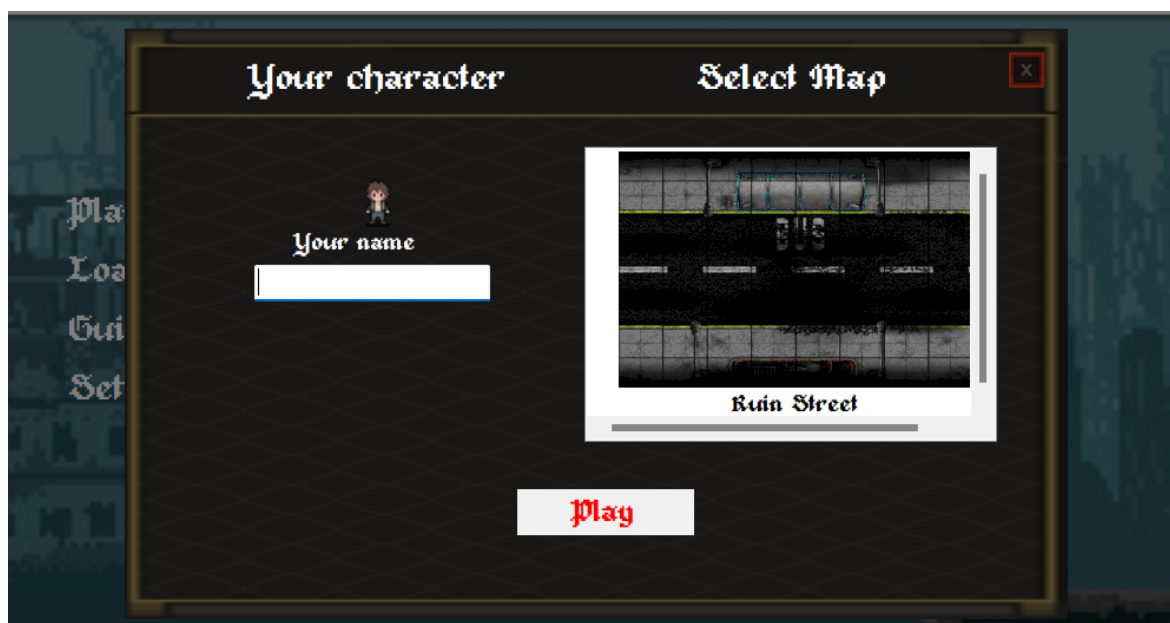
Hình 5-3. Cài đặt chế độ chơi

5.4. Kết nối cơ sở dữ liệu



Hình 5-4. Kết nối cơ sở dữ liệu

5.5. Chơi mới



Hình 5-5. Màn hình chơi mới

5.6. Chơi lại



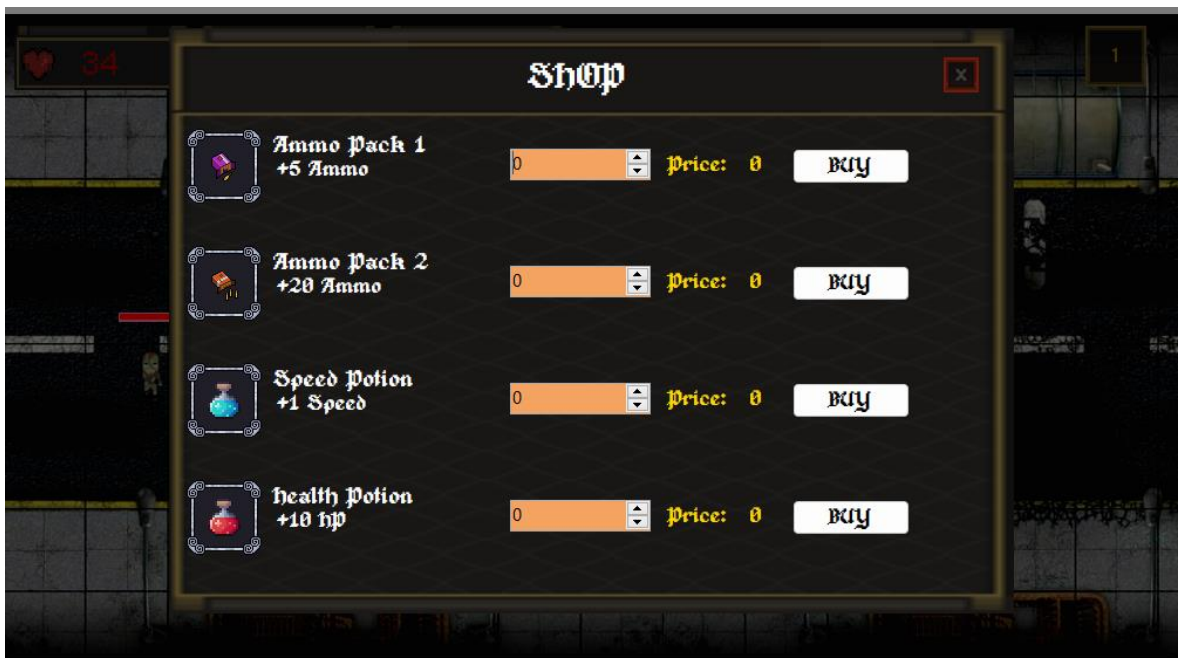
Hình 5-6. Màn hình chơi lại

5.7. Màn hình chính



Hình 5-7. Màn hình chơi chính

5.8. Cửa hàng



Hình 5-8. Cửa hàng

5.9. Nâng cấp vũ khí



Hình 5-9. Nâng cấp vũ khí

5.10. Hành trang



Hình 5-10. Hành trang

5.11. Chiến thắng



Hình 5-11. Màn chơi chiến thắng

5.12. Thất bại



Hình 5-12. Màn chơi thất bại

5.13. Tạm ngưng



Hình 5-13. Tạm ngưng màn chơi

Chương 6. KẾT LUẬN

6.1. Kết quả

Xây dựng được một trò chơi giải trí không chỉ là tạo ra nó mà còn là cả một quá trình khảo sát thực tế thị hiếu người dùng và những trò chơi hấp dẫn hơn những cái đã được tạo ra bởi những nhà sáng lập trước đó. Nó đòi hỏi chúng ta phải làm bằng cái tâm và thật sự tạo ra được một cái gì đó khác và lạ hơn so với những gì đã có. Chúng em cũng đã tự thực hiện được đam mê của chính mình, tự tạo ra một trò chơi theo phong cách mà mình yêu thích. Tạo ra một trò chơi không chỉ đơn giản là tạo ra một sự giải trí đơn thuần mà còn là gửi gắm vào đó những bài học bổ ích, tính nhân văn và văn hóa.

Đứng trước xu thế phát triển của công nghệ thông tin hiện nay có rất nhiều phần mềm làm game như COCOS2D, Unity, Unreal Developer Kit,... nhưng nhóm chúng em đã thực hiện đề tài này trên chính Windows Forms. Lựa chọn này rất thách thức và chúng em đã làm được và có thể củng cố được rất nhiều kiến thức về lập trình, cách thiết kế giao diện một cách hoà hòa và nắm được cách làm thế nào để xây dựng được một game từ những việc nhỏ nhất như lên ý tưởng, thiết kế giao diện cho đến những thứ to lớn hơn là viết code cho game hoạt động.



Hình 6-1. Windows Forms

6.2. Đánh giá

6.2.1. Những điều đạt được

- Xây dựng được Menu và các giao diện chính.
- Xây dựng được các tính năng trò chơi cần phải có.
- Có kết nối cơ sở dữ liệu.
- Hệ thống trò chơi hoạt động mượt, không bị xung đột dữ liệu.
- Có âm thanh sống động.

6.2.2. Những điều còn hạn chế

- Vòng chơi chưa được đa dạng.
- Việc xử lý và giải phóng dữ liệu chưa được hoàn toàn đảm bảo.

6.2.3. Hướng phát triển

Do là một trò chơi đầu tay của nhóm nên các tính năng vẫn còn bị hạn chế. Nhóm chúng em sẽ tiếp tục tạo nên các màn chơi mới, làm cho trò chơi trở nên đa dạng hơn và thú vị hơn. Bên cạnh đó sẽ thay đổi AI cho các đối tượng để thêm phần thú vị và kích thích người chơi.