**LỜI CẢM ƠN**

Em xin chân thành cảm ơn Khoa Công Nghệ Thông Tin, trường Đại Duy Tân đã tạo điều kiện tốt cho em thực hiện đề tài này.

Em xin chân thành cảm ơn Thầy Trịnh Hiệp Hòa, là người đã định hướng và giúp đỡ em trong suốt thời gian thực hiện đề tài. Trong quá trình thực hiện đề tài thầy đã tận tình chỉ dẫn, trao đổi giúp em giải quyết các vấn đề để hoàn thiện đề tài. Em cũng xin gửi lời cảm ơn sâu sắc đến quý thầy cô trong khoa đã tận tình giảng dạy và trang bị cho em vốn kiến thức vô cùng quí báu trong những năm học vừa qua. Cha, Mẹ, anh chị đã quan tâm, chăm sóc, động viên. Bạn bè đã ủng hộ, giúp đỡ chúng em trong những lúc khó khăn cũng như trong suốt thời gian học tập và nghiên cứu. Mặc dù em đã cố gắng hoàn thành luận văn trong phạm vi và khả năng cho phép, nhưng chắc chắn sẽ không tránh khỏi những thiếu sót, kính mong sự cảm thông và chỉ bảo của quý thầy cô và các bạn.

Đà Nẵng, ngày 16 tháng 3 năm 2025.

Sinh viện thực hiện

Đoàn Quang Đăng

LỜI CAM ĐOAN

Tôi xin cam đoan :

1. Những nội dung trong luận văn này là do tôi thực hiện dưới sự hướng dẫn trực tiếp của thầy Trịnh Hiệp Hòa.
2. Mọi tham khảo dùng trong luận văn đều được trích dẫn rõ ràng và trung thực tên tác giả, tên công trình, thời gian, địa điểm công bố.
3. Mọi sao chép không hợp lệ, vi phạm quy chế đào tạo, hay gian trá,  
   tôi xin chịu hoàn toàn trách nhiệm.

Đà Nẵng, ngày 22 tháng 3 năm 2025.

Sinh viện thực hiện

Đoàn Quang Đăng

# MỞ ĐẦU

**1. Lý do chọn đề tài**

**2. Mục đích và ý nghĩa của đề tài**

**3. Đối tượng và phạm vi nghiên cứu**

**4. Phương pháp nghiên cứu**

**5. Kết quả dự kiến**

**6. Bố cục đề tài**

*Đà Nẵng, ngày 16 tháng 3, năm 2025.*

# CHƯƠNG I

# TỔNG QUAN VỀ CÔNG CỤ VÀ MỖI TRƯỜNG PHÁT TRIỂN

## **GIỚI THIỆU NGÔN NGỮ LẬP TRÌNH C#**

### Lịch sử phát triển của ngôn ngữ lập trình C#

* C# (hay C sharp) là một ngôn ngữ lập trình đơn giản, được phát triển bởi đội ngũ kỹ sư của Microsoft vào năm 2000, trong đó người dẫn đầu là Anders Hejlsberg và Scott Wiltamuth.
* C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và nó được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.
* C# được thiết kế cho Common Language Infrastructure (CLI), mà gồm Executable Code và Runtime Environment, cho phép chúng ta sử dụng các ngôn ngữ high-level đa dạng trên các nền tảng và cấu trúc máy tính khác nhau.
* C# với sự hỗ trợ mạnh mẽ của .NET Framework giúp cho việc tạo một ứng dụng Windows Forms hay WPF (Windows Presentation Foundation), . . . trở nên rất dễ dàng.

### Một số đặc điểm nổi bậc của ngôn ngữ lập trình C#

* C# theo một hướng nào đó, là ngôn ngữ lập trình phản ánh trực tiếp nhất đến [.NET Framework](https://vi.wikipedia.org/wiki/.NET_Framework) mà tất cả các chương trình.NET chạy, và nó phụ thuộc mạnh mẽ vào framework này. Mọi dữ liệu cơ sở đều là đối tượng, được cấp phát và hủy bỏ bởi trình dọn rác Garbage-Collector (GC), và nhiều kiểu trừu tượng khác chẳng hạn như class, delegate, interface, exception... phản ánh rõ ràng những đặc trưng của.NET runtime.

So sánh với C và C++, ngôn ngữ này bị giới hạn và được nâng cao ở một vài đặc điểm nào đó, nhưng không bao gồm các giới hạn sau đây:

* Các con trỏ chỉ có thể được sử dụng trong chế độ không an toàn. Hầu hết các đối tượng được tham chiếu an toàn, và các phép tính đều được kiểm tra tràn bộ đệm. Các con trỏ chỉ được sử dụng để gọi các loại kiểu giá trị; còn những đối tượng thuộc bộ gom rác (*garbage-collector*) thì chỉ được gọi bằng cách tham chiếu.
* Các đối tượng không thể được giải phóng tường minh.
* Chỉ có đơn kế thừa, nhưng có thể cài đặt nhiều interface trừu tượng (abstract interfaces). Chức năng này làm đơn giản hóa sự thực thi của thời gian thực thi.
* C# thì an-toàn-kiểu (*typesafe*) hơn C++.
* Cú pháp khai báo mảng khác nhau ("int[] a = new int[5]" thay vì "int a[5]").
* Kiểu thứ tự được thay thế bằng tên miền không gian (*namespace*).
* C# không có tiêu bản.
* Có thêm Properties, các phương pháp có thể gọi các Properties để truy cập dữ liệu.
* Có reflection.
* Lập trình chức năng
* Hỗ trợ kiểu động

### 1.1.3 Đặc trưng của ngôn ngữ C#

* Đơn giản, dễ học: C# thừa hưởng ưu điểm của C++, Java, Visual Basic... và loại bỏ các điểm yếu, giúp cú pháp đơn giản, dễ tiếp cận cho người mới bắt đầu.
* Hiện đại, đa năng: C# phù hợp với nhiều lĩnh vực trong thời đại 4.0 như: phát triển web, ứng dụng mobile, game, AI, IoT, blockchain, microservices...
* Hướng đối tượng + chức năng: C# hỗ trợ lập trình hướng đối tượng mạnh mẽ, đồng thời tích hợp lập trình chức năng qua lambda, pattern matching, immutable data...
* Gõ tĩnh an toàn + hỗ trợ gõ động: Với kiểu gõ tĩnh, C# phát hiện lỗi sớm, an toàn kiểu. Kết hợp Visual Studio và IntelliCode giúp viết code nhanh và chính xác hơn.
* Ít từ khóa: C# có khoảng 80 từ khóa, giúp dễ học và dễ nhớ.
* Phổ biến toàn cầu: Theo TIOBE, PYPL và GitHub (2020), C# nằm trong top 5 ngôn ngữ phổ biến nhất. Cộng đồng hơn 6 triệu lập trình viên.
* Kết hợp chặt chẽ với .NET: C# là ngôn ngữ chủ lực của nền tảng .NET – hệ sinh thái mạnh mẽ do Microsoft phát triển và liên tục cải tiến.
* Mã nguồn mở, miễn phí, đa nền tảng: Chạy tốt trên Windows, Linux, MacOS với hiệu suất cao nhờ CLR. Phát triển phần mềm nhanh chóng.
* IDE mạnh mẽ + tài nguyên phong phú: Visual Studio và các công cụ khác như VS Code, Vim đều hỗ trợ C#. Tài liệu, hội thảo, cộng đồng hỗ trợ rất tốt.
* Cú pháp rõ ràng, logic nhất quán: Dễ đọc, dễ bảo trì, mở rộng – phù hợp cả với người mới học và lập trình viên chuyên nghiệp.

### Các ứng dụng C#

* Phát triển web [backend](https://vi.wikipedia.org/wiki/Front-end_v%C3%A0_back-end) ([ASP.NET](https://vi.wikipedia.org/wiki/ASP.NET) MVC, ASP.NET core, Web API,Graph API,gPRC, Blazor sevver,Uno platform, Mono)
* Phát triển web [front end](https://vi.wikipedia.org/wiki/Front-end_v%C3%A0_back-end) (Blazor WebAssembly,Uno platform)
* Phát triển desktop app (Winform, WPF, UWP,WinUI, Mono, Uno, MAUI,Blazor desktop...)
* Phát triển game 2D, 3D đa nền tảng ([Game engine](https://vi.wikipedia.org/wiki/Game_engine):[Unity](https://vi.wikipedia.org/wiki/Unity_(game_engine)), Monogame, Godot, Stride, CryEngine, Flax Engine, Evergine, NeoAxis, XNA ..)
* Phát triển thực tế ảo (VR), thực tế tăng cường(AR), thực tế hỗn hợp (MR) (HoloLens,Unity,CryEngine, Oculus quest..)
* Phát triển ứng dụng đồ họa 2D,3D đa nền tảng (2D: SkiaSharp, ImageSharp...; 3D: OpenTK, SharpDX, SharpVulkan, Vulkan.NET, Veldrid, Silk.NET, Helix Toolkit, Aspose..)
* Phát triển [mobile app](https://vi.wikipedia.org/wiki/%E1%BB%A8ng_d%E1%BB%A5ng_tr%C3%AAn_thi%E1%BA%BFt_b%E1%BB%8B_di_%C4%91%E1%BB%99ng), IOS native, [Android](https://vi.wikipedia.org/wiki/Android_(h%E1%BB%87_%C4%91i%E1%BB%81u_h%C3%A0nh)) native (Xamarin,MAUI,Uno platform)
* Phát triển [đám mây](https://vi.wikipedia.org/wiki/%C4%90i%E1%BB%87n_to%C3%A1n_%C4%91%C3%A1m_m%C3%A2y) ([Azure](https://azure.microsoft.com/en-us/),AWS,Google Cloud...)
* Học máy và [trí tuệ nhân tạo](https://vi.wikipedia.org/wiki/Tr%C3%AD_tu%E1%BB%87_nh%C3%A2n_t%E1%BA%A1o) ([ML.Net](https://dotnet.microsoft.com/apps/machinelearning-ai/ml-dotnet), TensorFlow, csiSharp..)
* Data science, bigdata (csiSharp, Apache Spark)
* [Blockchain](https://vi.wikipedia.org/wiki/Blockchain) ([NEO, Stratis](https://www.stratisplatform.com/))
* Microservices and containers
* Internet of thing (IoT,5G)
* Hệ thống nhúng ([Raspberry pi](https://vi.wikipedia.org/w/index.php?title=Raspberry_pi&action=edit&redlink=1), PLC)

## **GIỚI THIỆU ENGINE UNITY 6**

### Lịch sử phát triển

* Unity là một game engine đa nền tảng được phát triển bởi Unity Technologies, chủ yếu để phát triển video game cho máy tính, consoles và thiết bị di động. Ban đầu, Unity được công bố tại Apple's Worldwide Developers Conference vào năm 2005, chỉ hỗ trợ hệ điều hành Mac OS X. Kể từ đó, Unity đã mở rộng để hỗ trợ nhiều nền tảng khác như Windows, Linux, iOS, Android, PlayStation, Xbox, và Nintendo Switch.
* Các phiên bản chính của Unity đã lần lượt được phát hành:
* Unity 1.0 (2005): Hỗ trợ Mac OS X, tập trung vào phát triển game 3D.
* Unity 2.0 (2007): Mở rộng sang Windows, hỗ trợ trình duyệt web với Unity Web Player.
* Unity 3.0 (2010): Bổ sung hỗ trợ PlayStation 3, Xbox 360 và Android.
* Unity 4.0 (2012): Cải thiện đồ họa với DirectX 11, Mecanim animation và hỗ trợ Linux.
* Unity 5.0 (2015): Cập nhật Global Illumination, PBR (Physically-Based Rendering) và hỗ trợ WebGL.
* Unity 2017 - 2019: Giới thiệu Timeline, Cinemachine, và công nghệ DOTS (Data-Oriented Technology Stack).
* Unity 2020 - 2023: Nâng cấp HDRP, URP và hỗ trợ ray tracing.
* Unity 6 (2024): Tập trung vào tối ưu hiệu suất, hỗ trợ AI và tích hợp công nghệ hiện đại cho game và metaverse. Hiện nay, Unity là một trong những game engine phổ biến nhất, cạnh tranh trực tiếp với Unreal Engine.

### Các module trong Unity

Unity bao gồm nhiều module quan trọng giúp lập trình viên phát triển game dễ dàng:

* Unity Editor: Giao diện đồ họa chính để xây dựng, chỉnh sửa game.
* Rendering Engine: Hỗ trợ HDRP (High Definition Render Pipeline) và URP (Universal Render Pipeline) để tối ưu đồ họa.
* Physics Engine: Sử dụng NVIDIA PhysX và DOTS-based physics cho mô phỏng vật lý.
* Animation System: Mecanim hỗ trợ tạo và quản lý animation chuyên nghiệp.
* AI & Navigation: Công cụ điều hướng NavMesh và tích hợp AI giúp phát triển NPC thông minh hơn.
* Networking: Hỗ trợ Multiplayer với Netcode for GameObjects và Unity Transport.
* AR/VR Support: Tích hợp công cụ phát triển cho AR (Augmented Reality) và VR (Virtual Reality).
* Scripting: Sử dụng ngôn ngữ C# để phát triển gameplay và logic game.

### Các dự án nổi bật với Unity

Unity được sử dụng rộng rãi trong ngành công nghiệp game và nhiều lĩnh vực khác:

* Game nổi bật:
  + Hollow Knight: Game phiêu lưu hành động với đồ họa đẹp mắt.
  + Genshin Impact: Game thế giới mở với đồ họa ấn tượng và hệ thống chiến đấu hấp dẫn.
  + Among Us: Game multiplayer nổi tiếng với lối chơi sáng tạo.
  + Monument Valley: Game giải đố với thiết kế nghệ thuật đặc sắc.
* Ứng dụng ngoài game:
  + Kiến trúc & Xây dựng: Unity giúp mô phỏng thiết kế nội thất và công trình kiến trúc.
  + Giáo dục & Y tế: Sử dụng để tạo mô phỏng phẫu thuật, hướng dẫn thực tế ảo.
  + Phim & Hoạt hình: Công nghệ real-time rendering của Unity giúp tạo hiệu ứng phim nhanh chóng.
* Nhờ sự phát triển mạnh mẽ và cộng đồng hỗ trợ rộng lớn, Unity 6 tiếp tục là một nền tảng lý tưởng để phát triển game và ứng dụng tương tác đa lĩnh vực.

## **GIỚI THIỆU GAME TOWER DEFENSE**

1. **Lịch sử phát triển của Tower Defense.**

Lịch sử phát triển game tower defense bắt đầu từ những năm 1990 với các trò chơi chiến thuật thời gian thực. Những trò chơi này thường yêu cầu người chơi xây dựng và nâng cấp các tháp để ngăn chặn kẻ thù tiến vào căn cứ. Một trong những trò chơi đầu tiên có thể được coi là tiền thân của thể loại này là "Rampart".

Tuy nhiên, thể loại tower defense thực sự bùng nổ vào giữa những năm 2000 với sự ra đời của "Desktop Tower Defense". Trò chơi này đã thu hút hàng triệu người chơi và mở ra một kỷ nguyên mới cho thể loại game này. Kể từ đó, nhiều trò chơi tower defense khác đã được phát triển, bao gồm "Bloons Tower Defense" và "Kingdom Rush", mỗi trò chơi đều mang đến những cơ chế và chiến lược độc đáo, thu hút người chơi trên toàn thế giới.

1. **Tower Defense là gì? Thế nào thì được gọi là game Tower Defense?**

Tower Defense (TD) là một thể loại game chiến thuật, trong đó người chơi phải bảo vệ căn cứ của mình khỏi các đợt tấn công của kẻ địch bằng cách xây dựng và nâng cấp các trụ phòng thủ (tower). Những trụ này tự động tấn công kẻ địch khi chúng di chuyển qua bản đồ theo một tuyến đường cố định hoặc mở.

* + Đặc điểm của game Tower Defense:
    - *Xây dựng trụ phòng thủ:* Người chơi có thể đặt các trụ ở những vị trí chiến lược để ngăn chặn kẻ địch.
    - *Làn sóng kẻ địch (Wave):* Kẻ địch thường tấn công theo từng đợt, với độ khó tăng dần theo thời gian.
    - *Nâng cấp và chiến thuật:* Người chơi có thể nâng cấp trụ để tăng sức mạnh, tốc độ bắn hoặc thêm hiệu ứng đặc biệt. Việc chọn vị trí đặt trụ và cách nâng cấp ảnh hưởng lớn đến chiến thắng.
    - *Tài nguyên và quản lý:* Game thường yêu cầu người chơi thu thập và quản lý tài nguyên để xây dựng và nâng cấp trụ.
  + Thế nào thì được gọi là game Tower Defense?
    - Một trò chơi được gọi là Tower Defense khi có những yếu tố cốt lõi sau:
    - *Hệ thống phòng thủ bằng trụ:* Trò chơi tập trung vào việc xây dựng các trụ để ngăn chặn kẻ địch.
    - *Lối chơi chiến thuật:* Người chơi phải đưa ra quyết định hợp lý về việc đặt trụ và sử dụng tài nguyên để tối ưu hóa khả năng phòng thủ.
    - *Mục tiêu bảo vệ căn cứ:* Trò chơi kết thúc khi căn cứ của người chơi bị phá hủy hoặc khi người chơi đánh bại tất cả các làn sóng kẻ địch.
  + Các trò chơi nổi bật thuộc thể loại này bao gồm *Bloons TD*, *Plants vs. Zombies*, *Kingdom Rush* và nhiều tựa game khác.

## **GIỚI THIỆU MYSQL**

### 1.4.1 MySQL là gì?

* MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở, được sử dụng rộng rãi để lưu trữ và quản lý dữ liệu. Nó hỗ trợ SQL (Structured Query Language) để truy vấn và thao tác dữ liệu.

### 1.4.2 MySQL hoạt động như thế nào?

* Cấu trúc bảng (Table-Based): Dữ liệu được lưu trữ dưới dạng bảng có hàng và cột.
* Hệ thống truy vấn SQL: Sử dụng lệnh như SELECT, INSERT, UPDATE, DELETE để thao tác dữ liệu.
* Hỗ trợ ACID: Đảm bảo tính toàn vẹn dữ liệu qua Atomicity, Consistency, Isolation, Durability.
* Kết nối với ứng dụng: MySQL có thể kết nối với nhiều ngôn ngữ lập trình như C#, Java, Python, PHP.

### 1.4.3 Tại sao sử dụng MySQL?

* Hiệu suất cao: Tốc độ truy vấn nhanh, tối ưu cho web và ứng dụng lớn.
* Bảo mật tốt: Hỗ trợ mã hóa dữ liệu và quyền truy cập theo user.
* Hỗ trợ đa nền tảng: Chạy được trên Windows, Linux, macOS.
* Tích hợp dễ dàng: Hỗ trợ nhiều framework và công nghệ như Laravel, Django, .NET.
* Cộng đồng mạnh mẽ: Được sử dụng rộng rãi với tài liệu phong phú.

# CHƯƠNG II

# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## **KHẢO SÁT THỰC TRẠNG**

### 2.1.1. Tổng quan về thể loại Tower Defense

Tower Defense (TD) là một nhánh của thể loại game chiến thuật, trong đó người chơi phải xây dựng hệ thống phòng thủ để ngăn chặn các đợt tấn công của kẻ địch. Dòng game này phổ biến trên cả nền tảng PC, console và mobile nhờ lối chơi đơn giản nhưng mang tính chiến thuật cao.

Hiện nay, có nhiều tựa game Tower Defense nổi bật như Bloons TD, Kingdom Rush, Plants vs. Zombies, Defense Grid, v.v. Những trò chơi này thường có các yếu tố chính như:

* Xây dựng và nâng cấp trụ phòng thủ để chặn kẻ địch.
* Hệ thống tài nguyên giúp người chơi xây dựng chiến lược hợp lý.
* Cấp độ khó tăng dần qua từng màn chơi.
* Cơ chế quái vật theo từng đợt (Wave System) tạo sự thử thách cho người chơi.

### 2.1.2. Thực trạng game Tower Defense trên thị trường

Mặc dù dòng game Tower Defense đã tồn tại từ lâu, nhưng vẫn có sự đổi mới về lối chơi và đồ họa để phù hợp với xu hướng hiện đại. Một số xu hướng hiện nay bao gồm:

* Kết hợp với yếu tố RPG: Một số game cho phép người chơi nâng cấp nhân vật hoặc có các anh hùng hỗ trợ chiến đấu.
* Lối chơi đa dạng hơn: Bên cạnh cơ chế truyền thống, một số game cho phép di chuyển tháp hoặc thay đổi chiến thuật linh hoạt hơn.
* Đồ họa và âm thanh nâng cao: Sự cải tiến về mặt hình ảnh giúp game trở nên hấp dẫn hơn, đặc biệt trên nền tảng mobile.
* Mô hình kinh doanh miễn phí (Free-to-Play): Nhiều game TD hiện nay áp dụng mô hình này, kết hợp với quảng cáo hoặc mua vật phẩm trong game.

### 2.1.3. Hạn chế của các game Tower Defense hiện nay

Mặc dù thể loại Tower Defense có nhiều ưu điểm nhưng vẫn còn một số hạn chế:

* Thiếu sự sáng tạo: Nhiều game có lối chơi lặp lại, không có điểm khác biệt rõ ràng.
* Độ khó mất cân bằng: Một số game quá dễ hoặc quá khó, không có sự điều chỉnh phù hợp theo kỹ năng người chơi.
* Phụ thuộc vào giao dịch trong game: Một số trò chơi Free-to-Play buộc người chơi phải mua vật phẩm để có trải nghiệm tốt hơn, gây mất cân bằng.

### 2.1.4. Hướng phát triển của game Tower Defense

Với những xu hướng và hạn chế trên, các game Tower Defense trong tương lai có thể tập trung vào:

* Tạo ra cơ chế gameplay mới mẻ, kết hợp yếu tố chiến thuật khác để tăng sự đa dạng.
* Phát triển AI thông minh hơn, giúp quái vật có thể phản ứng linh hoạt thay vì chỉ đi theo một lộ trình cố định.
* Cải thiện trải nghiệm người chơi, như hệ thống nhiệm vụ đa dạng, đồ họa sống động và khả năng chơi nhiều người.

## **MỘT SỐ TỰA GAME TOWER DEFENSE NỔI BẬT TRÊN THỊ TRƯỜNG**

### 2.2.1. Bloons TD Series (Bloons TD 6)



*Hình 2.2.3.a Bloons TD Series*

* Nhà phát triển: Ninja Kiwi
* Nền tảng: PC, Mobile
* Mô tả: Đây là một trong những series Tower Defense phổ biến nhất với lối chơi chiến thuật hấp dẫn. Người chơi đặt các trụ phòng thủ là những con khỉ có khả năng đặc biệt để ngăn chặn bong bóng (bloons) đi qua bản đồ. Bloons TD 6 được đánh giá cao nhờ đồ họa sinh động, hệ thống nâng cấp phong phú và chế độ co-op nhiều người chơi.

### 2.2.2. Kingdom Rush Series



*Hình 2.2.3.b Kingdom Rush Series*

* Nhà phát triển: Ironhide Game Studio
* Nền tảng: PC, Mobile
* Mô tả: Kingdom Rush là một trong những tựa game Tower Defense thành công nhất, nổi bật với lối chơi chiến thuật đặc sắc và phong cách đồ họa vẽ tay. Game có nhiều loại tháp phòng thủ, anh hùng hỗ trợ chiến đấu, và hệ thống kẻ địch đa dạng.

### 2.2.3. Plants vs. Zombies



*Hình 2.2.3.c Plant vs Zombies*

* Nhà phát triển: PopCap Games (sau này thuộc EA)
* Nền tảng: PC, Mobile, Console
* Mô tả: Dù không hoàn toàn giống với Tower Defense truyền thống, nhưng Plants vs. Zombies vẫn có cơ chế phòng thủ bằng việc đặt cây cối để chống lại zombie. Lối chơi sáng tạo, vui nhộn cùng với sự đa dạng về chiến thuật đã giúp game trở thành một trong những tựa game được yêu thích nhất trên toàn thế giới.

## **TÀI LIỆU THIẾT KẾ GAME 2D TOWER DEFENSE**

### 2.3.1 Game Mechanics (Cơ chế game):

### 2.3.1.1 Xây dựng trụ phòng thủ (Tower Building)

* Người chơi có thể đặt trụ phòng thủ (tower) lên những vị trí hợp lệ trên bản đồ.
* Mỗi trụ có các loại chỉ số khác nhau:
  + Trụ sát thương vật lý

### 2.3.1.2 Nâng cấp trụ (Tower Upgrade)

* Trụ có thể nâng cấp để tăng:
* Tầm bắn.
* Sát thương.
* Tốc độ bắn.

### 2.3.1.3 Sóng kẻ địch (Wave System)

* Kẻ địch xuất hiện theo từng đợt.
* Mỗi đợt có độ khó tăng dần: số lượng, tốc độ, giáp, kỹ năng đặc biệt.
* Một số loại kẻ địch có khả năng đặc biệt:
  + Nhanh — di chuyển tốc độ cao.
  + Trâu bò — máu và giáp cao.
  + Boss — Sát thương lớn, máu khủng.

### 2.3.1.4 Mục tiêu chiến thắng (Win/Lose Condition)

* Người chơi phải ngăn chặn kẻ địch đi hết đường và tiến vào căn cứ.
* Mỗi kẻ địch vượt qua sẽ trừ máu căn cứ.
* Game kết thúc khi máu căn cứ về 0 hoặc tất cả kẻ địch bị tiêu diệt

### 2.3.2 Gameplay (Trải nghiệm người chơi)

### 2.3.2.1 Lối chơi cơ bản

* Người chơi quan sát đường đi của kẻ địch.
* Quyết định vị trí đặt trụ hợp lý, nâng cấp và bố trí chiến thuật sao cho tiêu diệt kẻ địch hiệu quả.
* Phản ứng linh hoạt với từng loại kẻ địch qua mỗi đợt.

### 2.3.2.2 Cấp độ thử thách

* Cấp độ càng cao, số lượng và tốc độ của kẻ địch càng lớn.
* Các yếu tố tăng dần:
  + Nhiều loại kẻ địch hơn.
  + Nhiều đợt Boss.
  + Môi trường thay đổi.
  + Cấp độ AI của kẻ địch (né đạn, chia nhóm, tấn công trụ).

### 2.3.2.3 Chiến thuật & Ra quyết định

* Cân bằng giữa xây mới, nâng cấp, và tiết kiệm tiền cho các tình huống bất ngờ.
* Xác định trụ ưu tiên: sát thương cao, làm chậm, buff...
* Đọc vị chiến thuật của AI kẻ địch để đặt trụ phù hợp.

## **CÁC ĐỐI TƯỢNG TƯƠNG TÁC VỚI HỆ THỐNG**

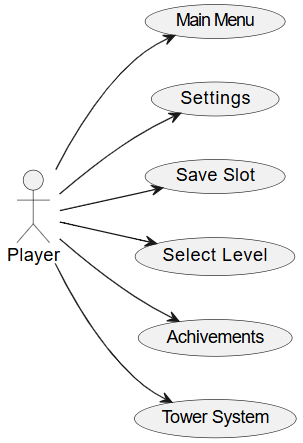
Hệ thống có các tác nhân: người dùng chưa đăng ký, người dùng đã đăng ký và người quản trị. Vai trò của các tác nhân được thể hiện trong bảng như sau:

|  |  |  |
| --- | --- | --- |
| STT | Tên tác nhân | Mô tả |
| 1 | Người chơi | **Tương tác với Main Menu**:   * Bắt đầu trò chơi (Start Game). * Xem thông tin nhóm phát triển (Credits). * Cài đặt âm thanh trong phần Settings.   **Chọn Save Slot**:   * Chọn một trong ba slot để lưu tiến trình hoặc tiếp tục chơi.   **Tương tác với bản đồ**:   * Chọn level để chơi. * Xem thông tin về các thực thể trong game (Book). * Kiểm tra thành tích (Achievements).   **Chơi trong Level**:   * Bắt đầu màn chơi. * Xây dựng, nâng cấp, hoặc bán tháp phòng thủ. * Điều chỉnh âm lượng hoặc thoát về Main Menu thông qua Settings. |

*Bảng 2.4 Các tác nhân của hệ thống*

## **SƠ ĐỒ USE CASE SỬ DỤNG (USE CASE DIAGRAM)**

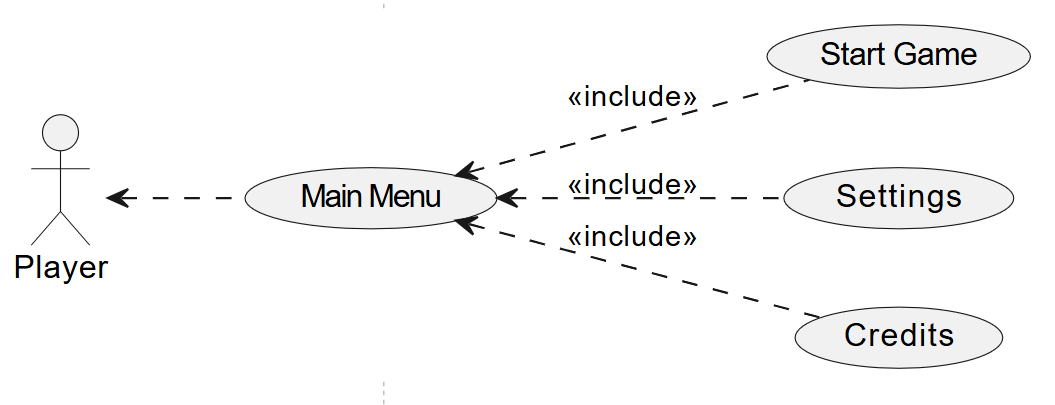
* + 1. **Use case tổng quát**
* Biểu đồ Use case người chơi (Player)



*Hình 2.5.1.a Use case người chơi*

* + 1. **Use case chi tiết**

#### *Use case Main Menu:*



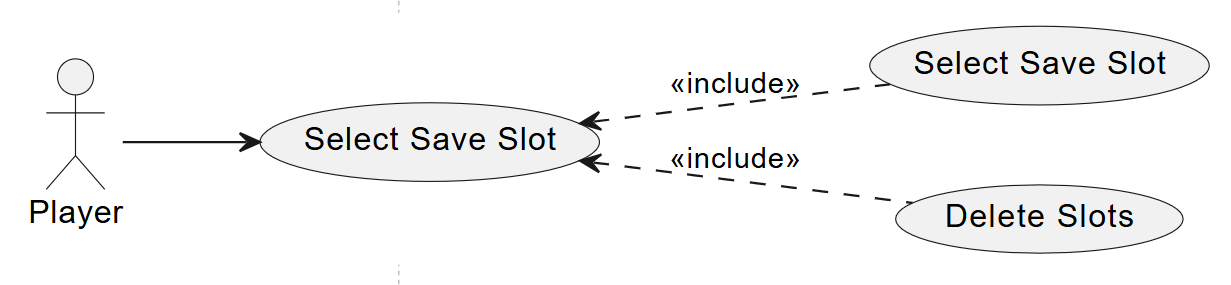
*Hình 2.5.2.1 Use case Main menu*

* Mục đích: Cho phép người chơi tương tác với main menu
* Tác nhân, mô tả chung:
  + Tác nhân: Người chơi
  + Mô tả chung: Mô tả việc người chơi thực hiện tương tác với main meu
* Luồng sự kiện chính:

|  |  |  |
| --- | --- | --- |
| STT | Hành động của tác nhân | Phản ứng của hệ thống |
| 1 | Người dùng chọn vào Start game. | Hiển thị màn hình save slots |
| 2 | Người dùng chọn vào Settings | Hiển thị một panel tắt bật âm thanh |
| 3 | Người dùng chọn vào Credits | Hiển thị credits của game |

*Bảng 2.5.2.1 Luồng sự kiện chính use case main menu*

#### *2.5.2.2 Use case Save Slots*



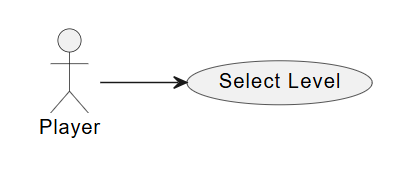
*Hình 2.5.2.2 Use case Main menu*

* Mục đích: Cho phép người chơi quản lý và tải dữ liệu lưu trữ
* Tác nhân, mô tả chung:
  + Tác nhân: Người chơi
* Mô tả chung: Cho phép người chơi quản lý và tải dữ liệu lưu trữ
* Luồng sự kiện chính:

|  |  |  |
| --- | --- | --- |
| STT | Hành động của tác nhân | Phản ứng của hệ thống |
| 1 | Người dùng chọn vào Save Slots. | khi Player chọn save slots thì sẽ vào save system check xem data có tồn tại hay ko, khi data tồn tại thì sẽ load data lên cho actor, còn khi data rỗng thì sẽ create data mới và load data lên cho Player. |
| 2 | Người dùng chọn vào Delete Slots. | Delete thì xóa đi data rồi trả lại data rỗng cho Player. |

*Bảng 2.5.2.2 Luồng sự kiện chính use case Save Slots*

#### *2.5.2.3 Use case Select Level*



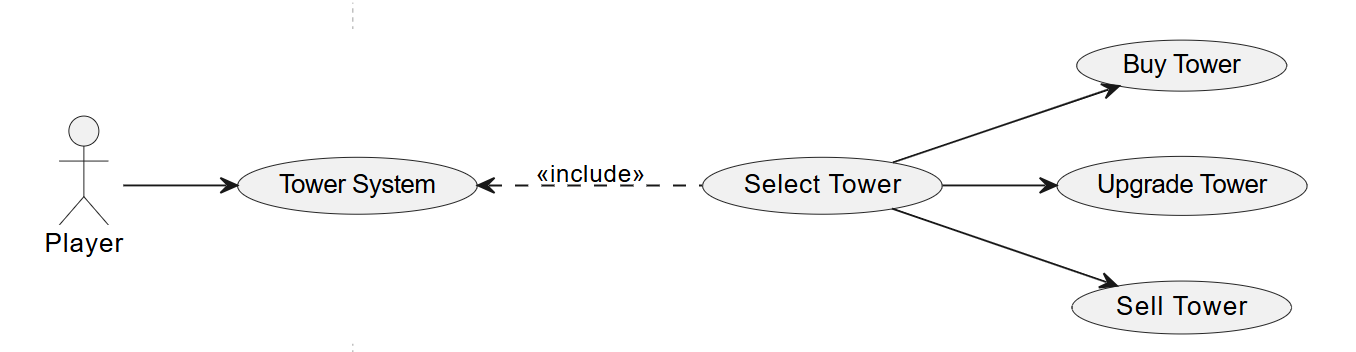
*Hình 2.5.2.3 Use case Select Level*

* **Mục đích:** Cho phép người chơi chọn Level
* **Tác nhân, mô tả chung:**
  + **Tác nhân:** Người chơi
* **Mô tả chung:** Cho phép người chơi chọn Level
* **Luồng sự kiện chính:**

|  |  |  |
| --- | --- | --- |
| STT | Hành động của tác nhân | Phản ứng của hệ thống |
| 1 | Người dùng chọn Level. | Hiển thị màn hình game của Level mà người chơi đã chọn. |

*Bảng 2.5.2.3 Luồng sự kiện chính use case chọn Level*

#### *2.5.2.4 Use case Tower System*



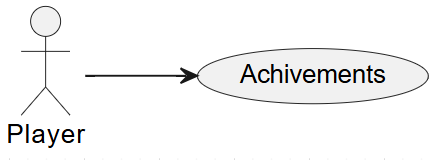
*Hình 2.5.2.4 Use case Select Level*

* Mục đích: Cho phép người chơi sử dụng hệ thống tower của game
* Tác nhân, mô tả chung:
  + Tác nhân: Người chơi
* Mô tả chung: Cho phép người chơi chọn tower và mua, bán hoặc nâng cấp
* Luồng sự kiện chính:

|  |  |  |
| --- | --- | --- |
| STT | Hành động của tác nhân | Phản ứng của hệ thống |
| 1 | Người dùng chọn Select Tower | Hiển thị thông tin của các Tower |
| 2 | Người dùng chọn Buy Tower | Mua Tower mà người dùng select trước đó |
| 3 | Người dùng chọn Upgrade Tower | Nâng cấp Tower |
| 4 | Người dùng chọn Sell Tower | Bán Tower và quay lại việc select tower |

*Bảng 2.5.2.4 Luồng sự kiện chính use case chọn Level*

#### *2.5.2.5 Use case Achievements*



*Hình 2.5.2.5 Use case Achievements*

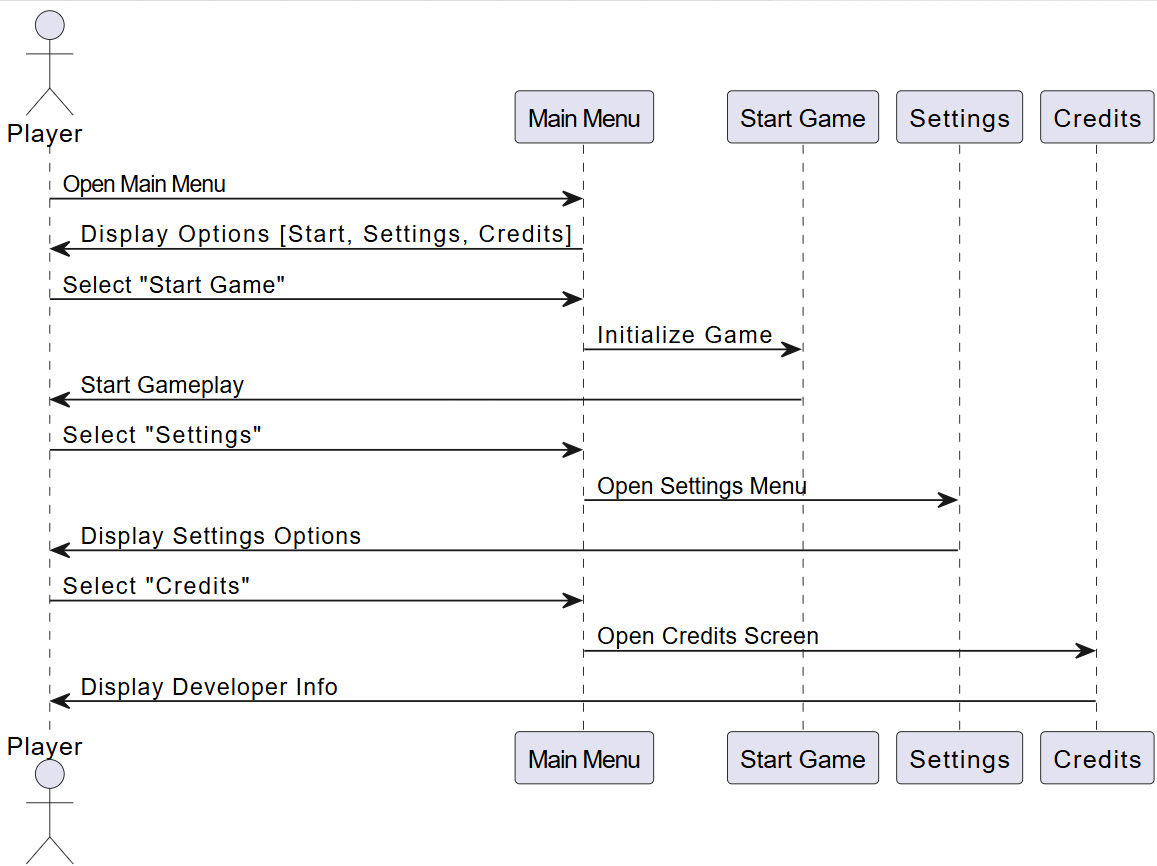
* Mục đích: Cho phép người chơi xem Achievements
* Tác nhân, mô tả chung:
  + Tác nhân: Người chơi
* Mô tả chung: Cho phép người chơi xem Achievements
* Luồng sự kiện chính:

|  |  |  |
| --- | --- | --- |
| STT | Hành động của tác nhân | Phản ứng của hệ thống |
| 1 | Người dùng chọn Level | Hiển thị màn hình xem Achievements người chơi có thể xem các thành tựu mà mình đã đạt được trong game |

*Bảng 2.5.2.5 Luồng sự kiện chính use case chọn Level*

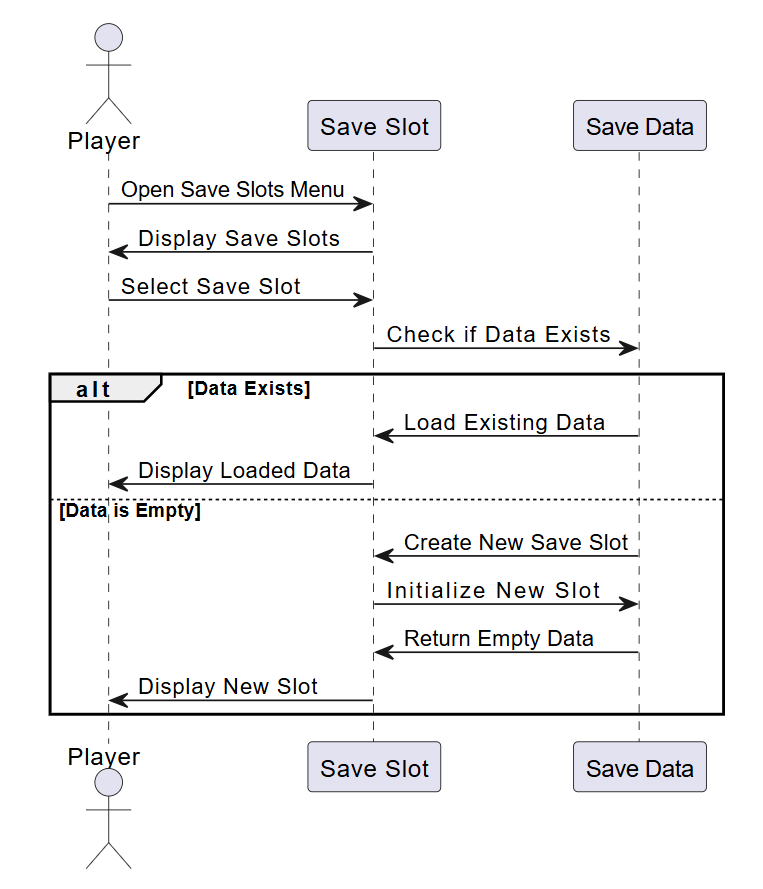
## **SƠ ĐỒ TUẦN TỰ (SEQUENSE DIAGRAM)**

### Sơ đồ tuần tự Main meu



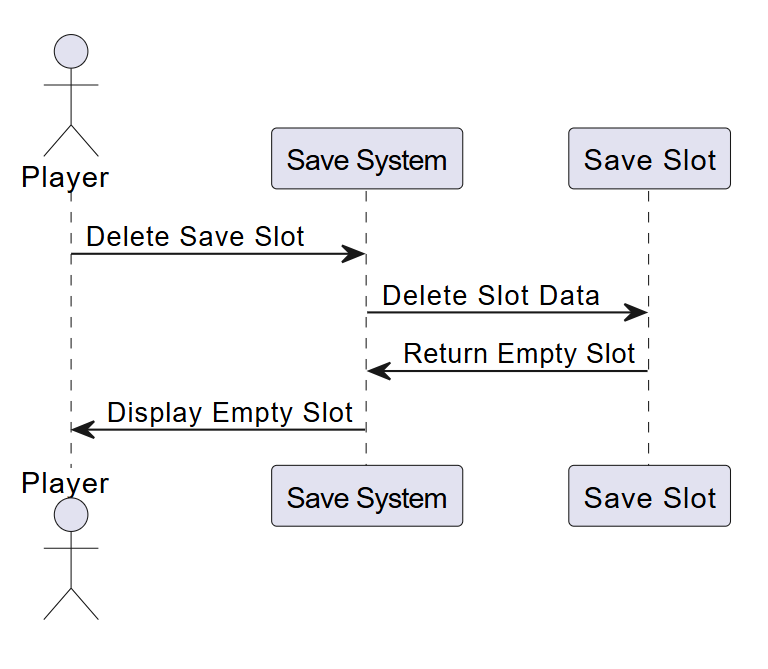
*Hình 2.6.1 Sơ đồ tuần tự Main menu*

### Sơ đồ tuần tự Save slots



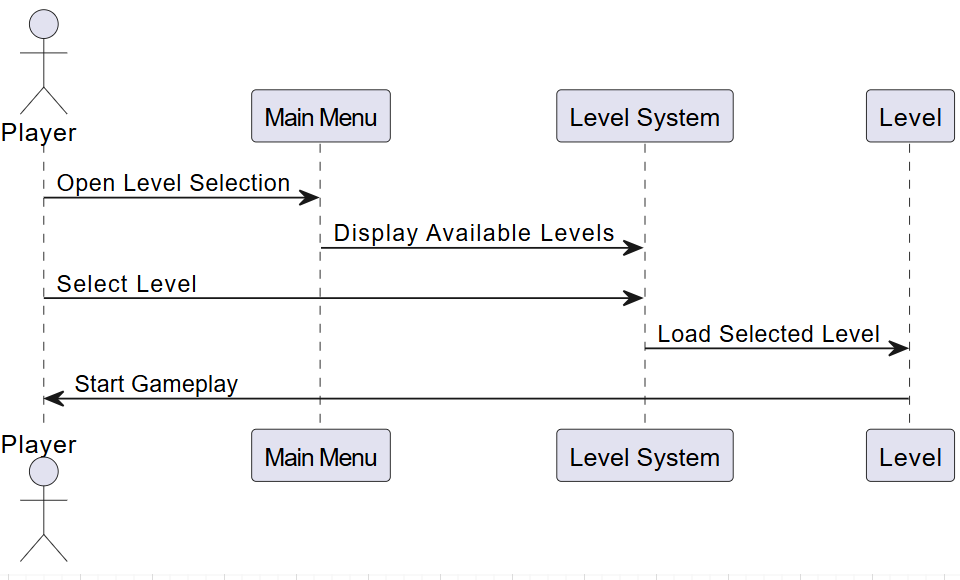
*Hình 2.6.2 Sơ đồ tuần tự Save slots*

### Sơ đồ tuần tự Delete Slots



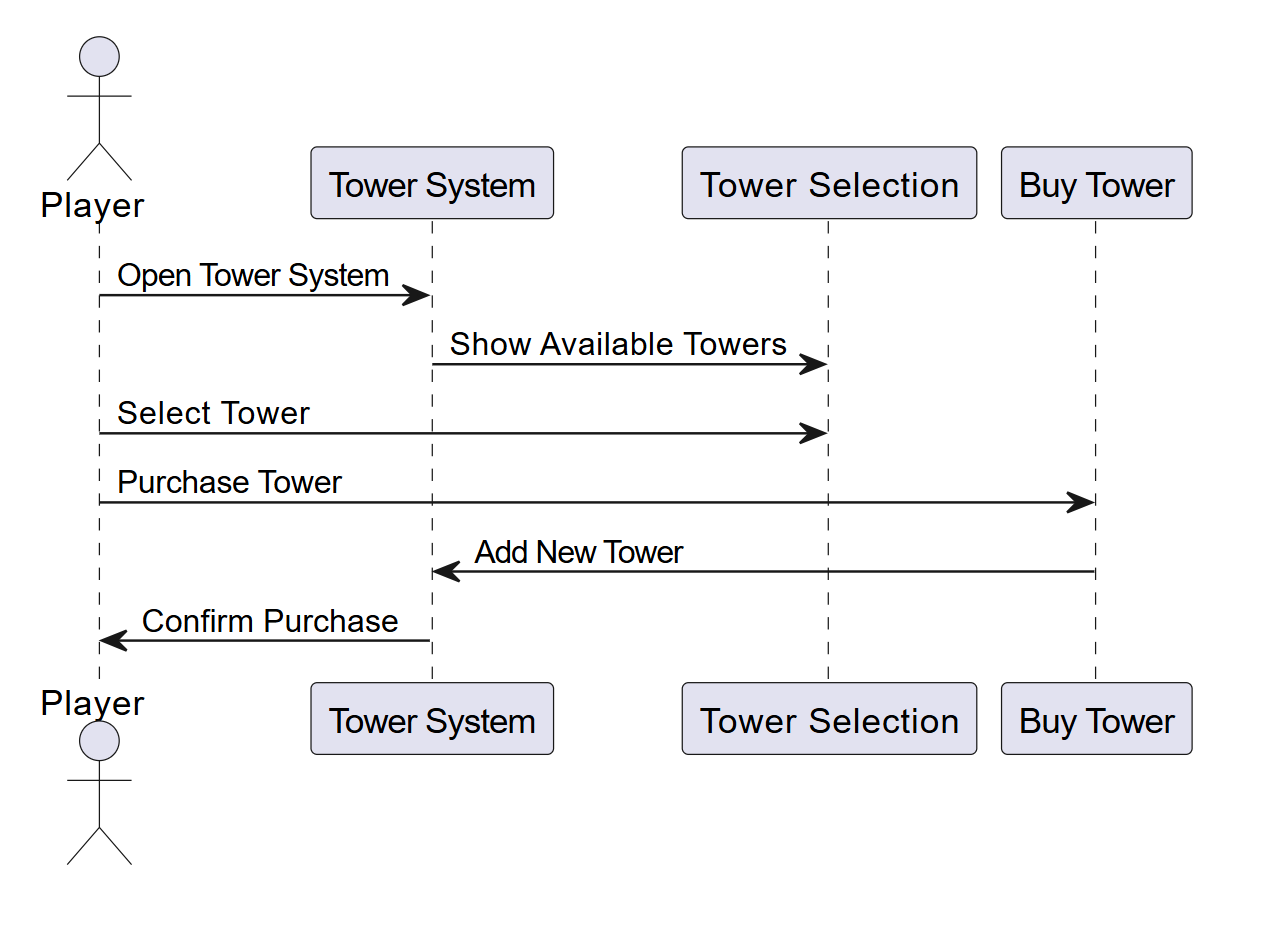
*Hình 2.6.3 Sơ đồ tuần tự Delete Slots*

### Sơ đồ tuần tự Select level



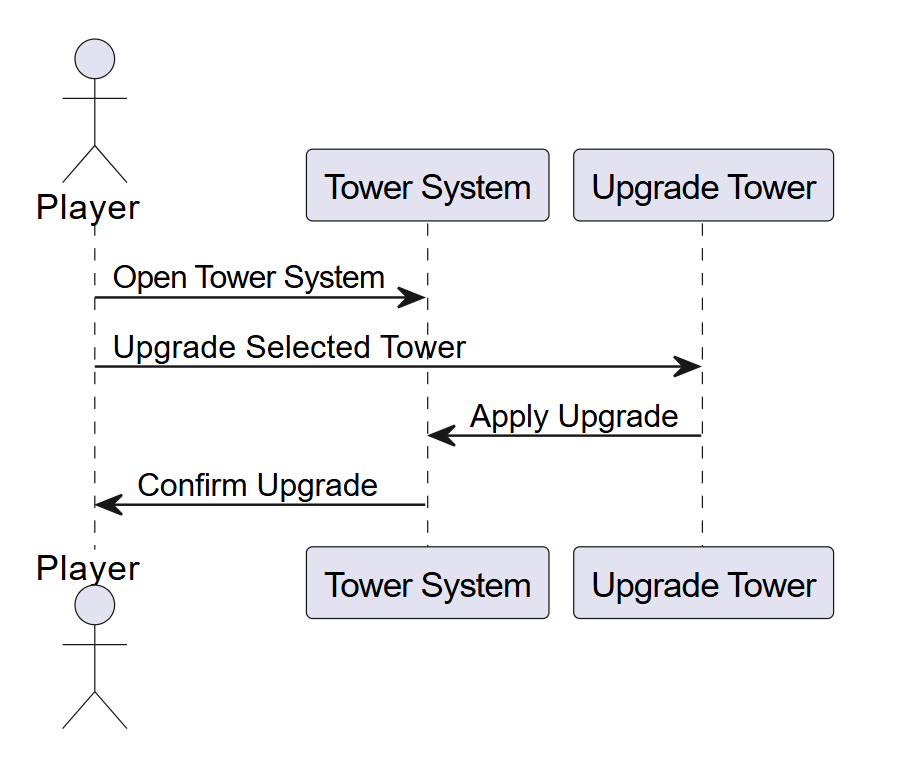
*Hình 2.6.4 Sơ đồ tuần tự Select level*

### Sơ đồ tuần tự Buy Tower



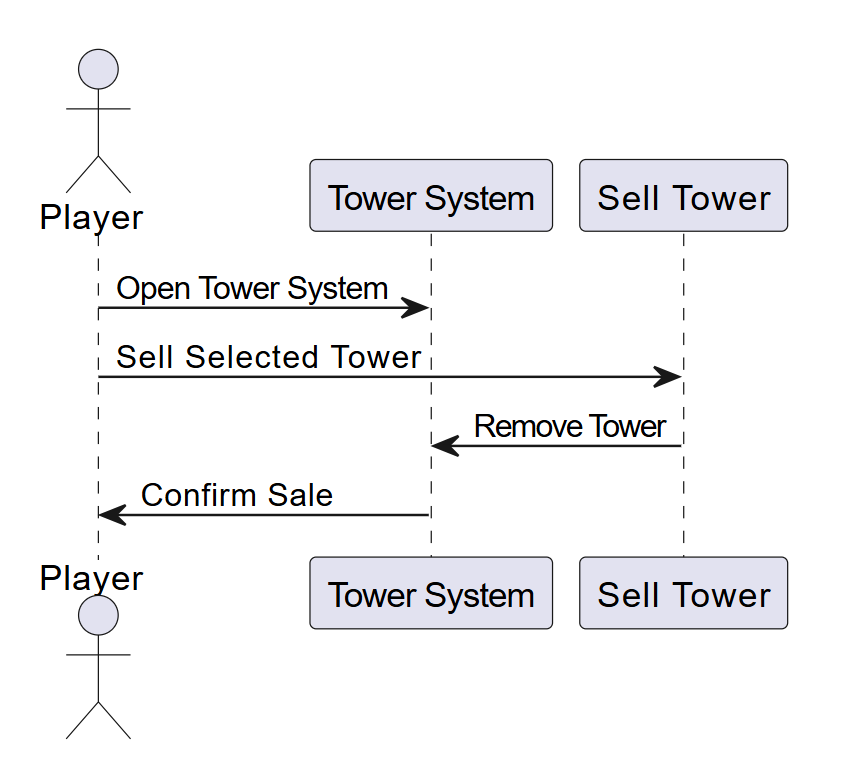
*Hình 2.6.5 Sơ đồ tuần tự Buy Tower*

### Sơ đồ tuần tự Upgrade Tower



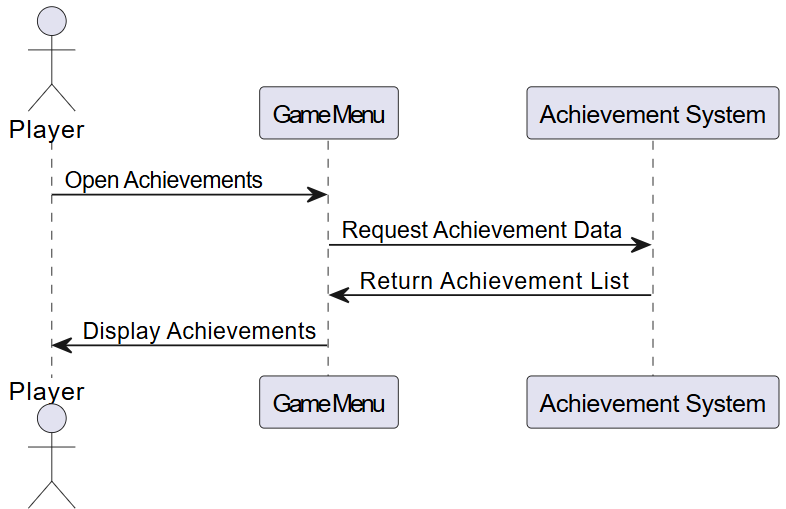
*Hình 2.6.6 Sơ đồ tuần tự UpgradeSystem*

### Sơ đồ tuần tự Sell Tower



*Hình 2.6.7 Sơ đồ tuần tự Sell System*

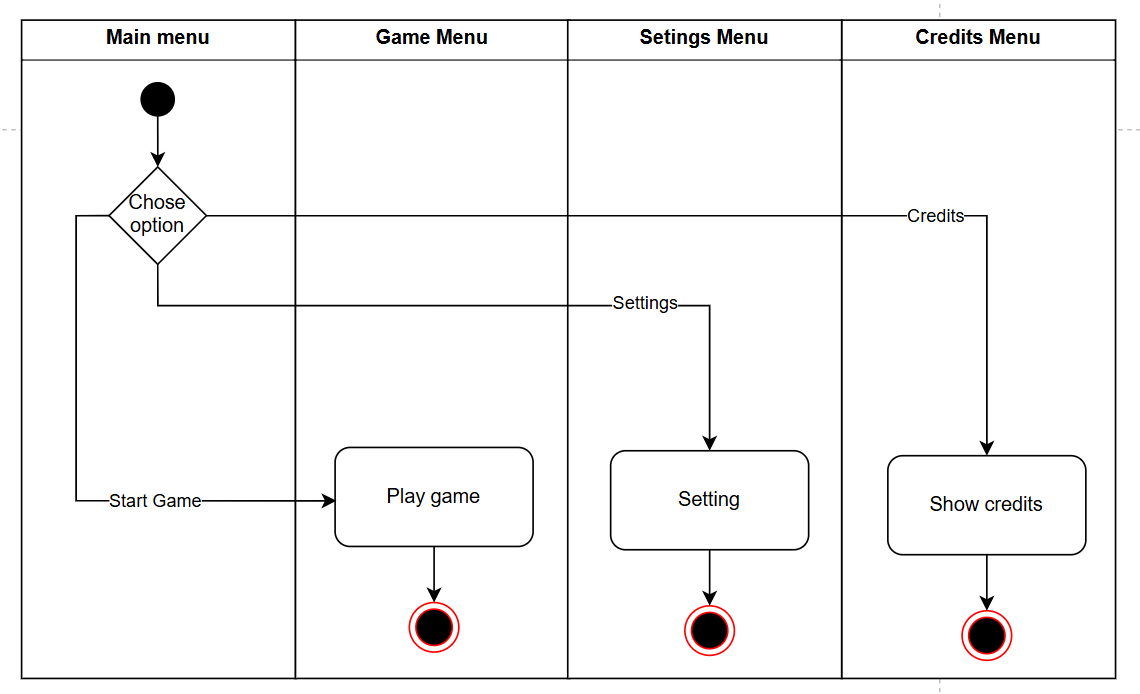
### Sơ đồ tuần tự Achievements:



*Hình 2.6.8 Sơ đồ tuần tự Achievements*

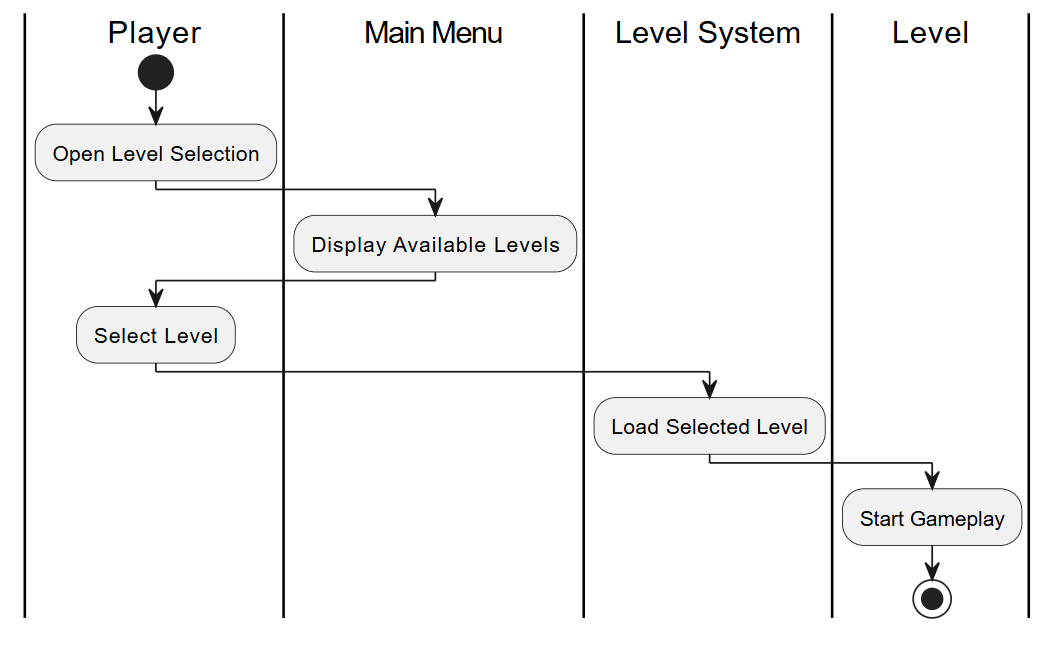
## **SƠ ĐỒ HOẠT ĐỘNG (ACTIVITY DIAGRAM)**

### Sơ đồ hoạt động đăng nhập



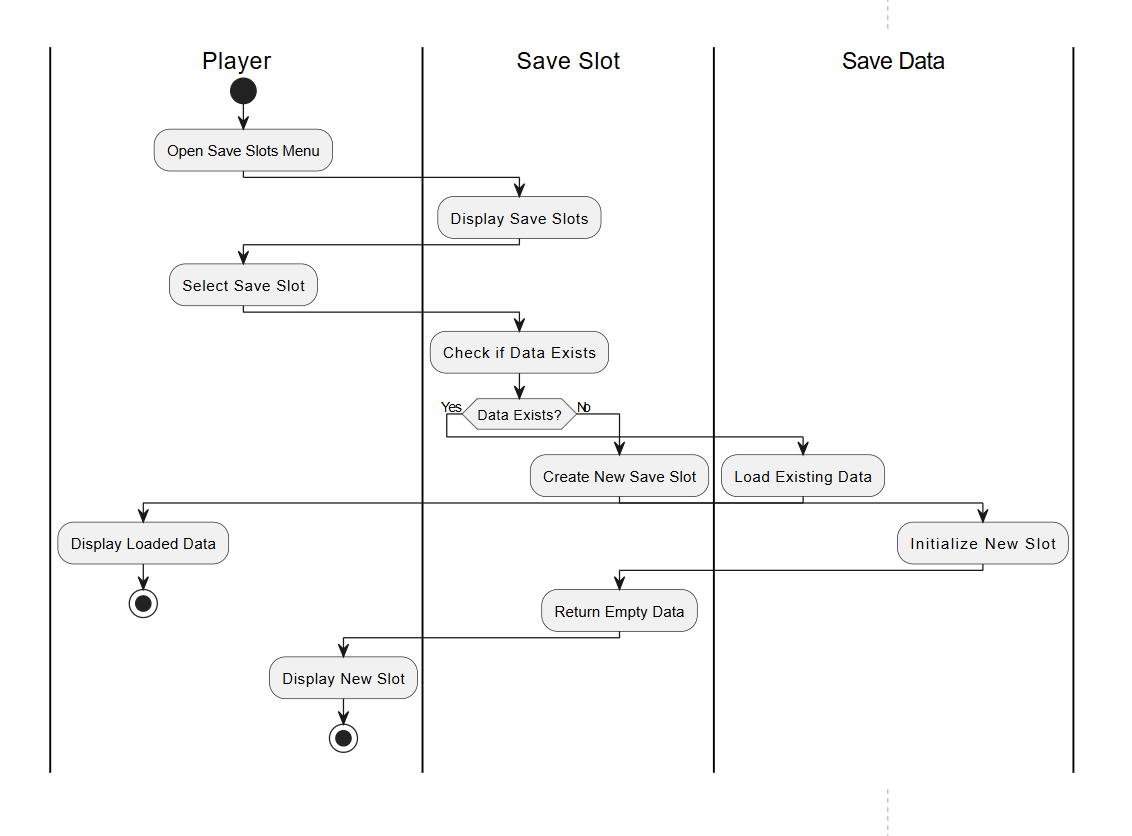
*Hình 2.7.1 Sơ đồ hoạt động main menu*

### Sơ đồ hoạt động Select Level



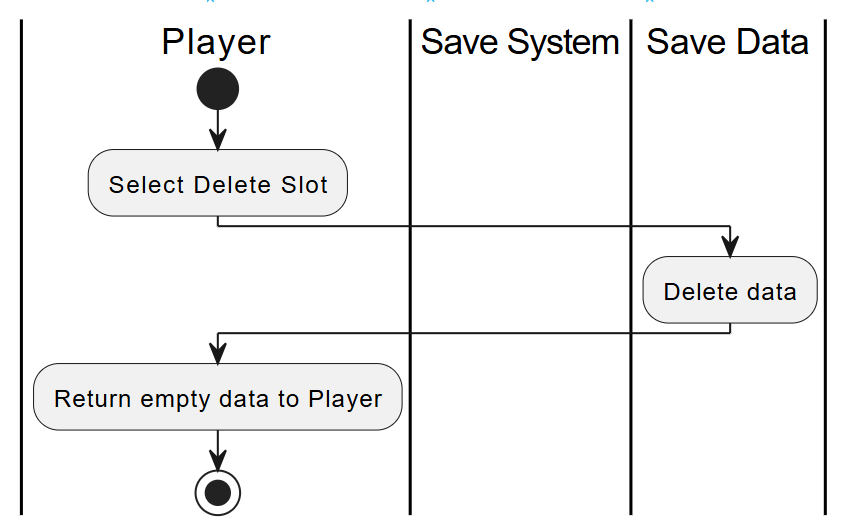
*Hình 2.7.2 Sơ đồ hoạt động Select Level*

### Sơ đồ hoạt động Save Slots



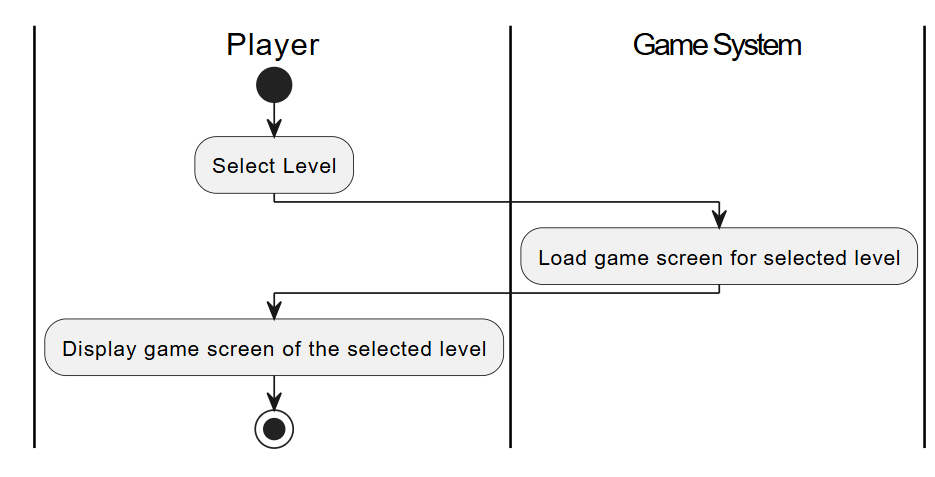
*Hình 2.7.3 Sơ đồ hoạt động Save slots*

### Sơ đồ hoạt động Delete Slots

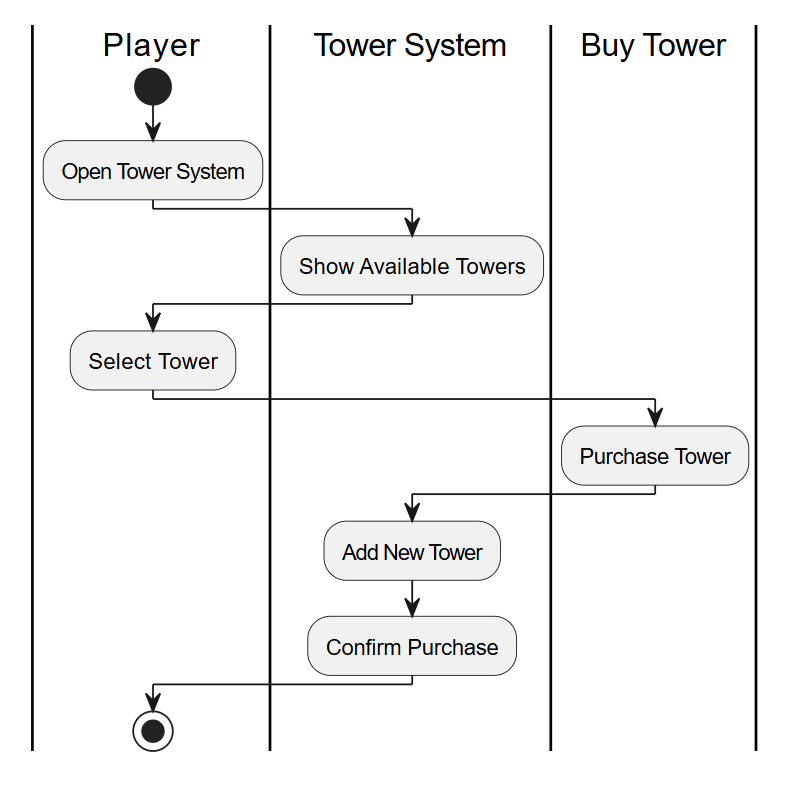


*Hình 2.7.4 Sơ đồ hoạt động Delete Slots*

### Sơ đồ hoạt động Select Level

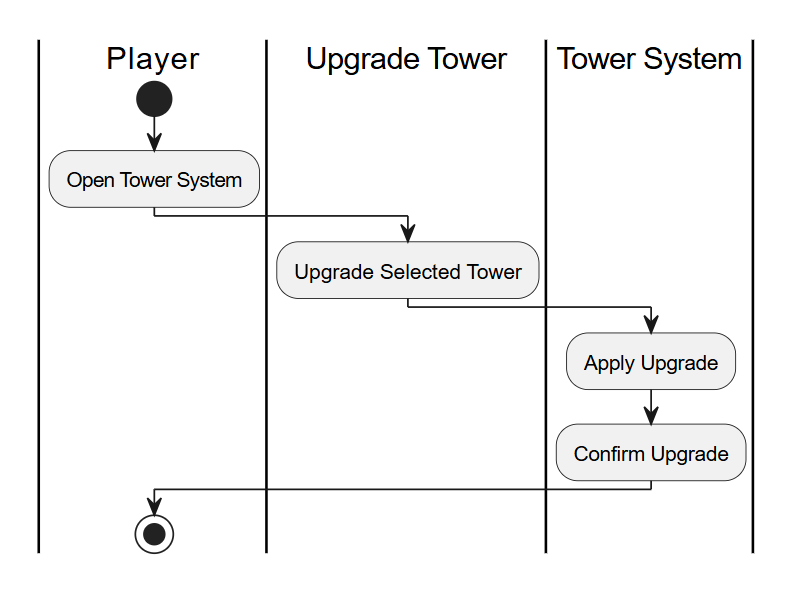
 *Hình 2.7.5 Sơ đồ hoạt động Select Level*

### Sơ đồ hoạt động Buy Tower



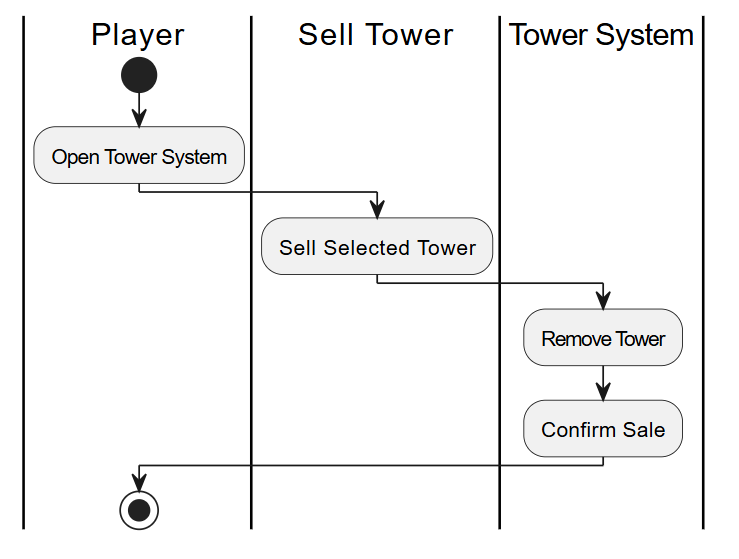
*Hình 2.7.6 Sơ đồ hoạt động Sell Tower*

### Sơ đồ hoạt động Upgrade Tower



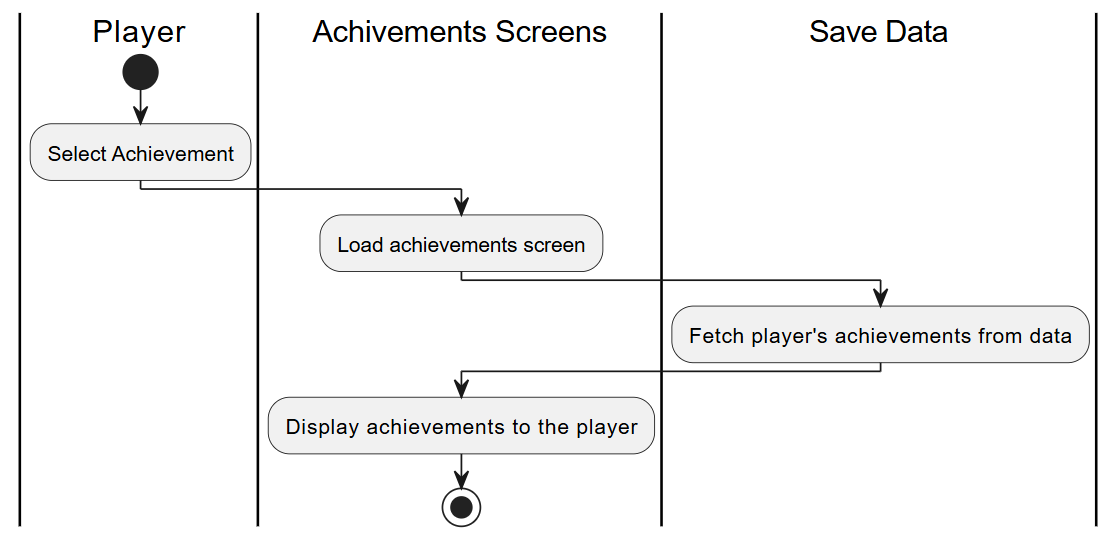
*Hình 2.7.7 Sơ đồ hoạt động Upgrade Tower*

### Sơ đồ hoạt động Sell Tower



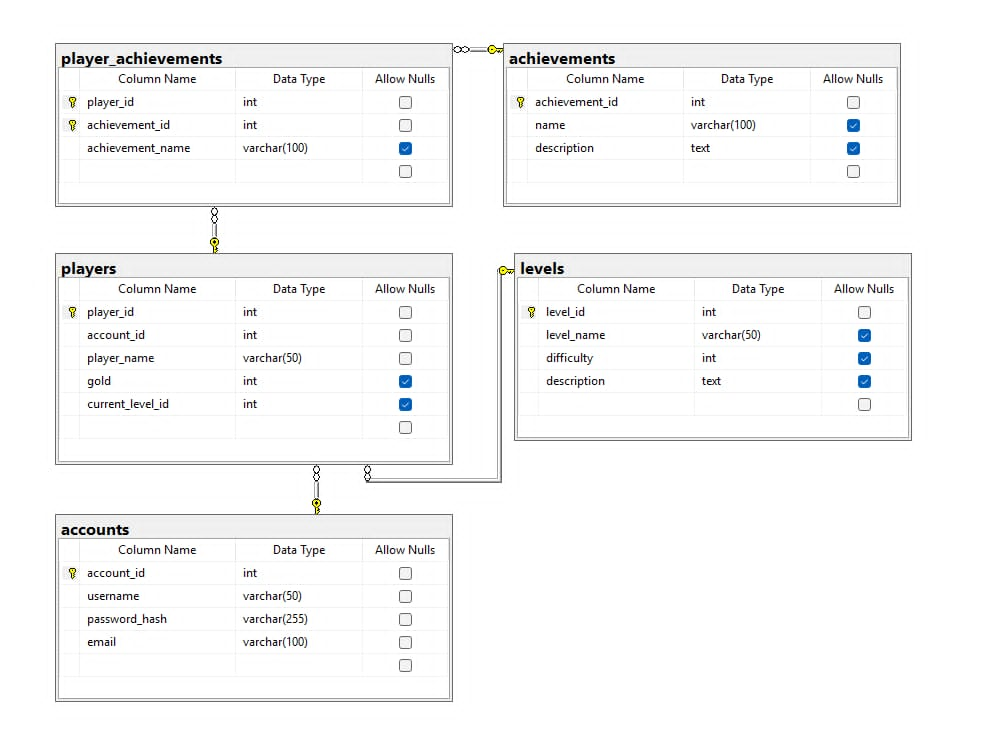
*Hình 2.7.8 Sơ đồ hoạt động Sell Tower*

### Sơ đồ hoạt động Achievements



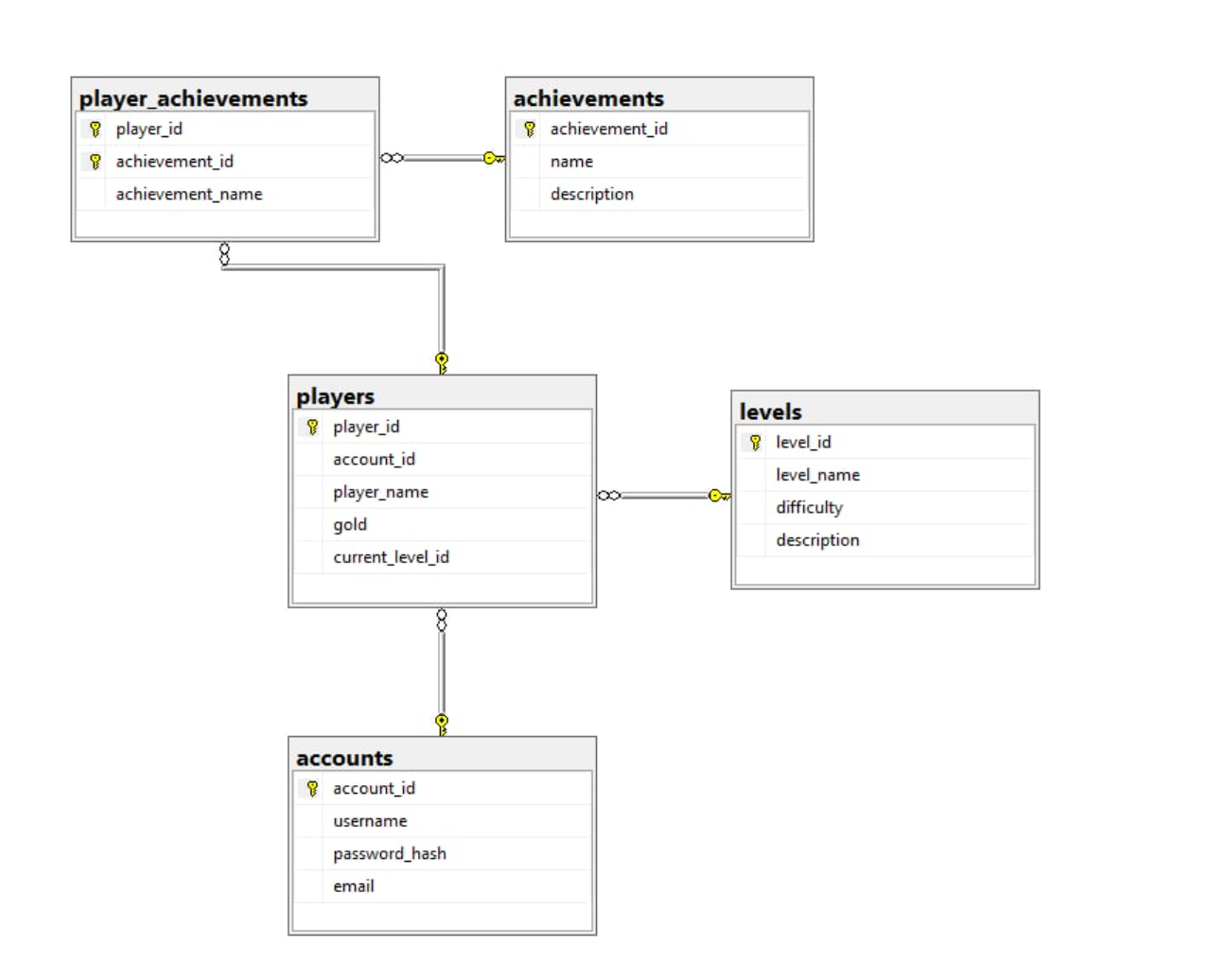
*Hình 2.7.9 Sơ đồ hoạt động Achievements*

## **SƠ ĐỒ LỚP (CLASS DIAGRAM)**



## **THIẾT KẾ CƠ SỞ DỮ LIỆU**

### Mô hình quan hệ



### Mô hình vật lý

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Extra** | **Link to** |
| account\_id | INT | No | PRIMARY KEY |  |
| username | VARCHAR(50) | No | UNIQUE |  |
| password\_hash | VARCHAR(255) | No |  |  |
| email | VARCHAR(100) | No | UNIQUE |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Extra** | **Link to** |
| level\_id | INT | No | PRIMARY KEY |  |
| level\_name | VARCHAR(50) | Yes |  |  |
| difficulty | INT | Yes | CHECK (1–10) |  |
| description | TEXT | Yes |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Extra** | **Link to** |
| player\_id | INT | No | PRIMARY KEY |  |
| account\_id | INT | No |  | accounts.account\_id |
| player\_name | VARCHAR(50) | No |  |  |
| gold | INT | Yes | DEFAULT 0 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Extra** | **Link to** |
| achievement\_id | INT | No | PRIMARY KEY |  |
| name | VARCHAR(100) | No |  |  |
| description | TEXT | Yes |  |  |

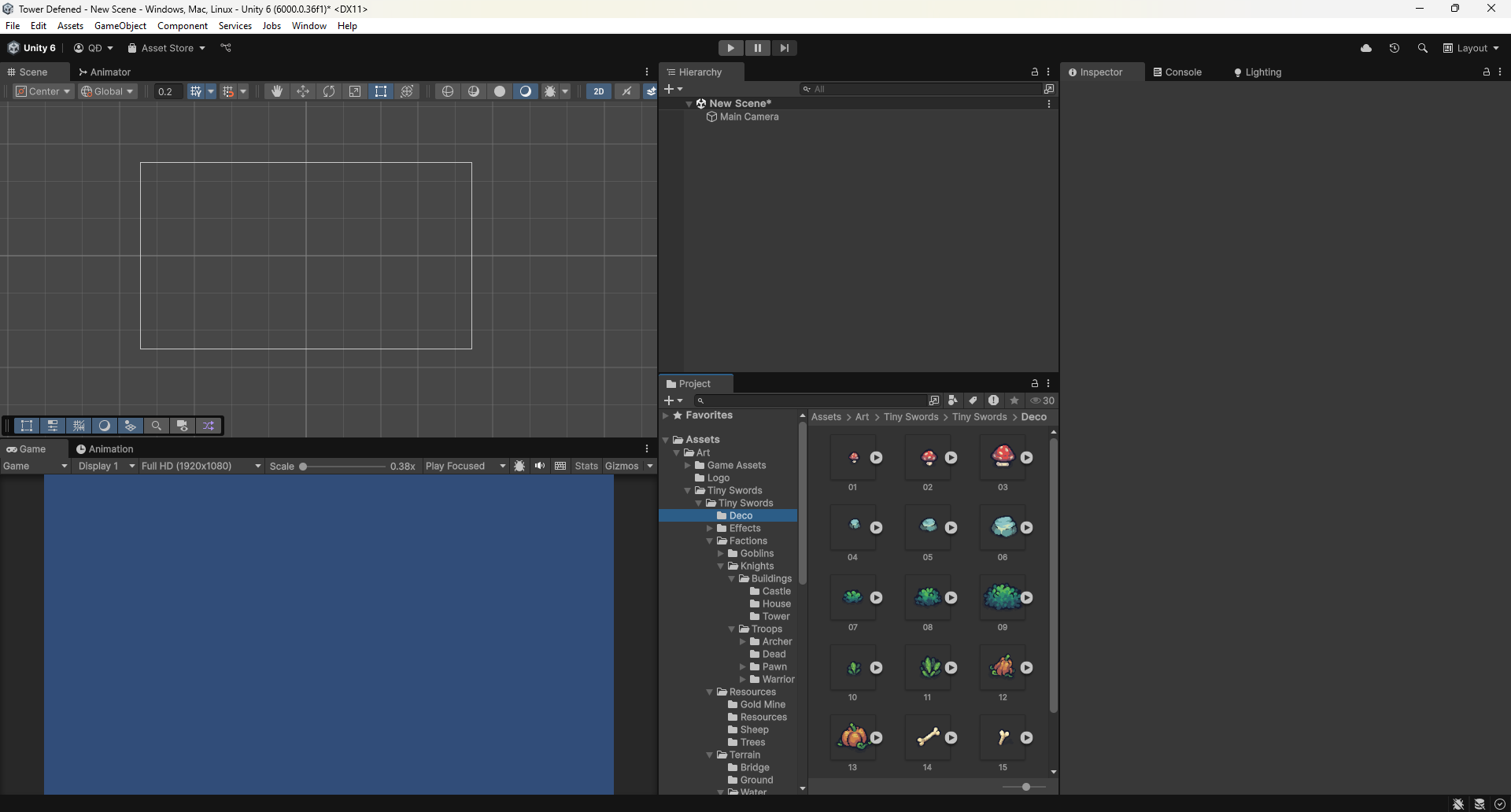
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Extra** | **Link to** |
| player\_id | INT | No | PRIMARY KEY (\*) | players.player\_id |
| achievement\_id | INT | No | PRIMARY KEY (\*) | achievements.achievement\_id |
| achievement\_name | VARCHAR(100) | Yes |  |  |

# CHƯƠNG III

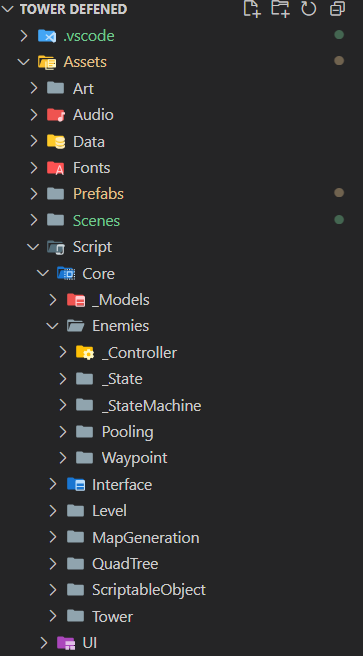
# CÀI ĐẶT VÀ KIỂM THỬ

## **CÀI ĐẶT**

1. **Khởi tạo dự án**

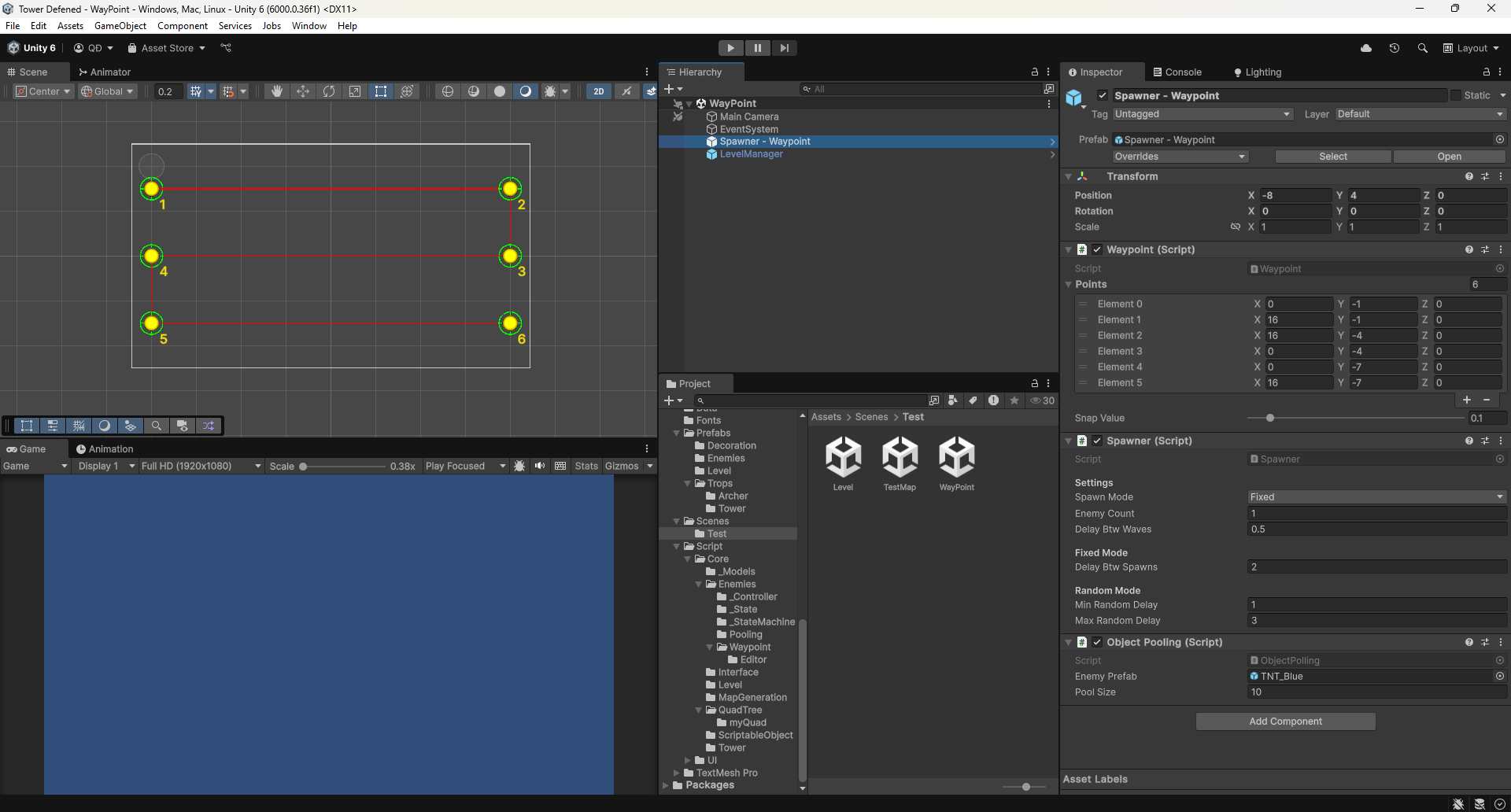
****

*Hình 3.1.1.a Khởi tạo dự án và thêm resource*

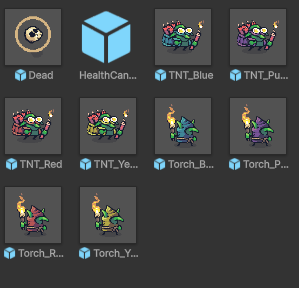
**

*Hình 3.1.1.b Cấu trúc thư mục*

1. **Enemy System**



*Hình 3.1.2.a Tạo Waypoint Editor System*

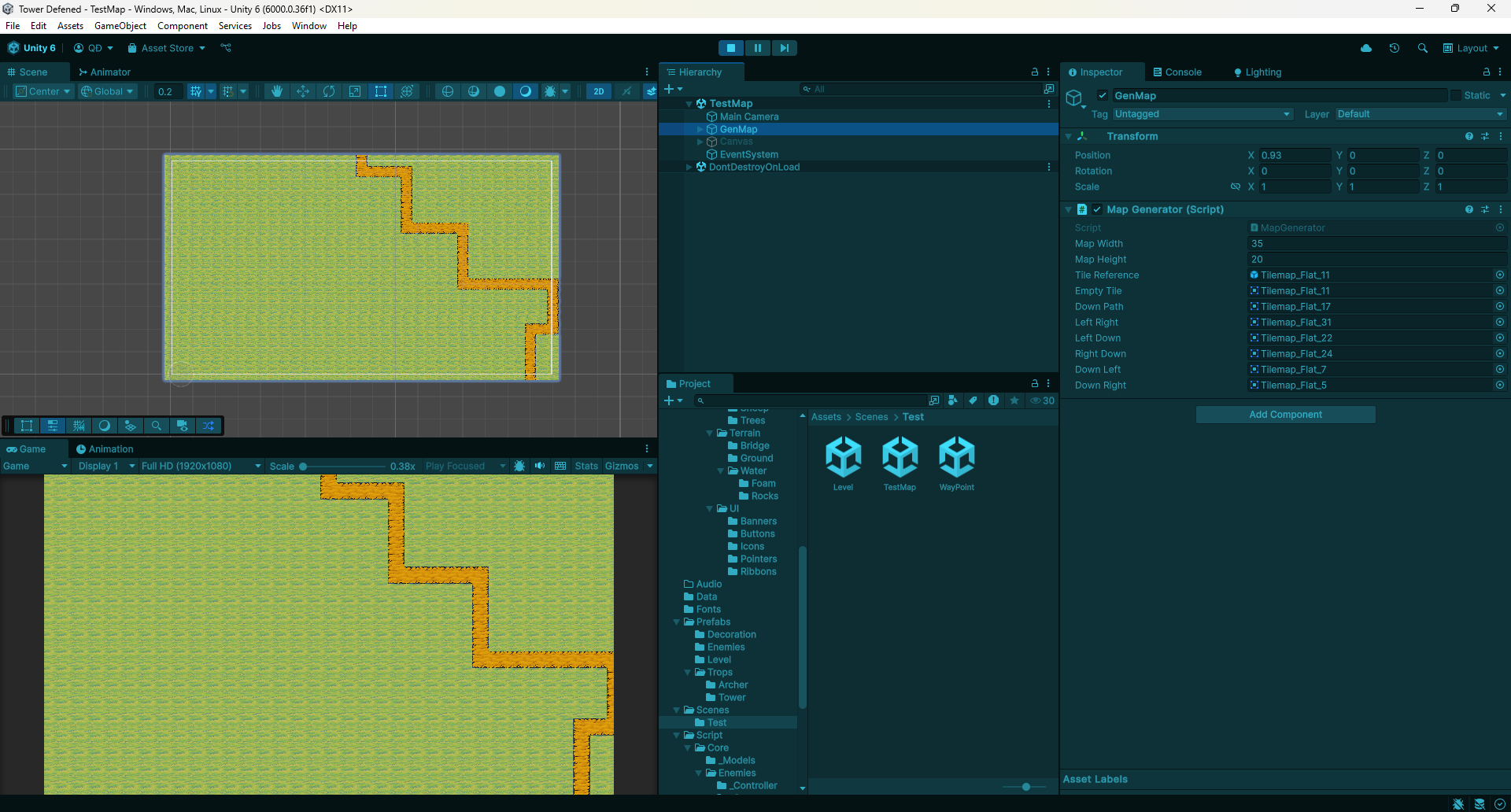
**

*Hình 3.1.2.b Tạo enemy prefabs*

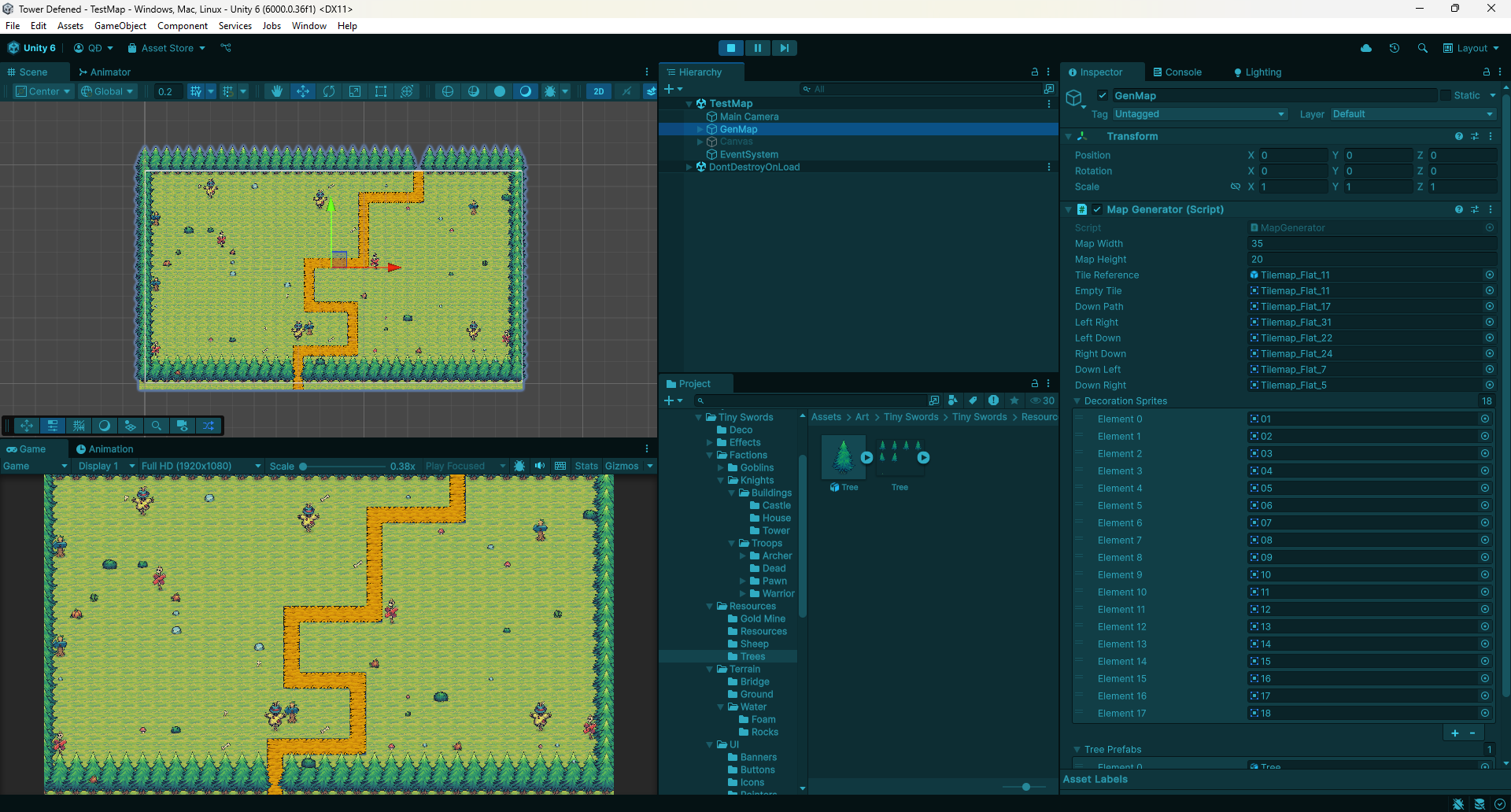
**

*Hình 3.1.2.c Tạo hệ thống cho enemy (Máu, tốc độ, Animation)*

1. **Level**

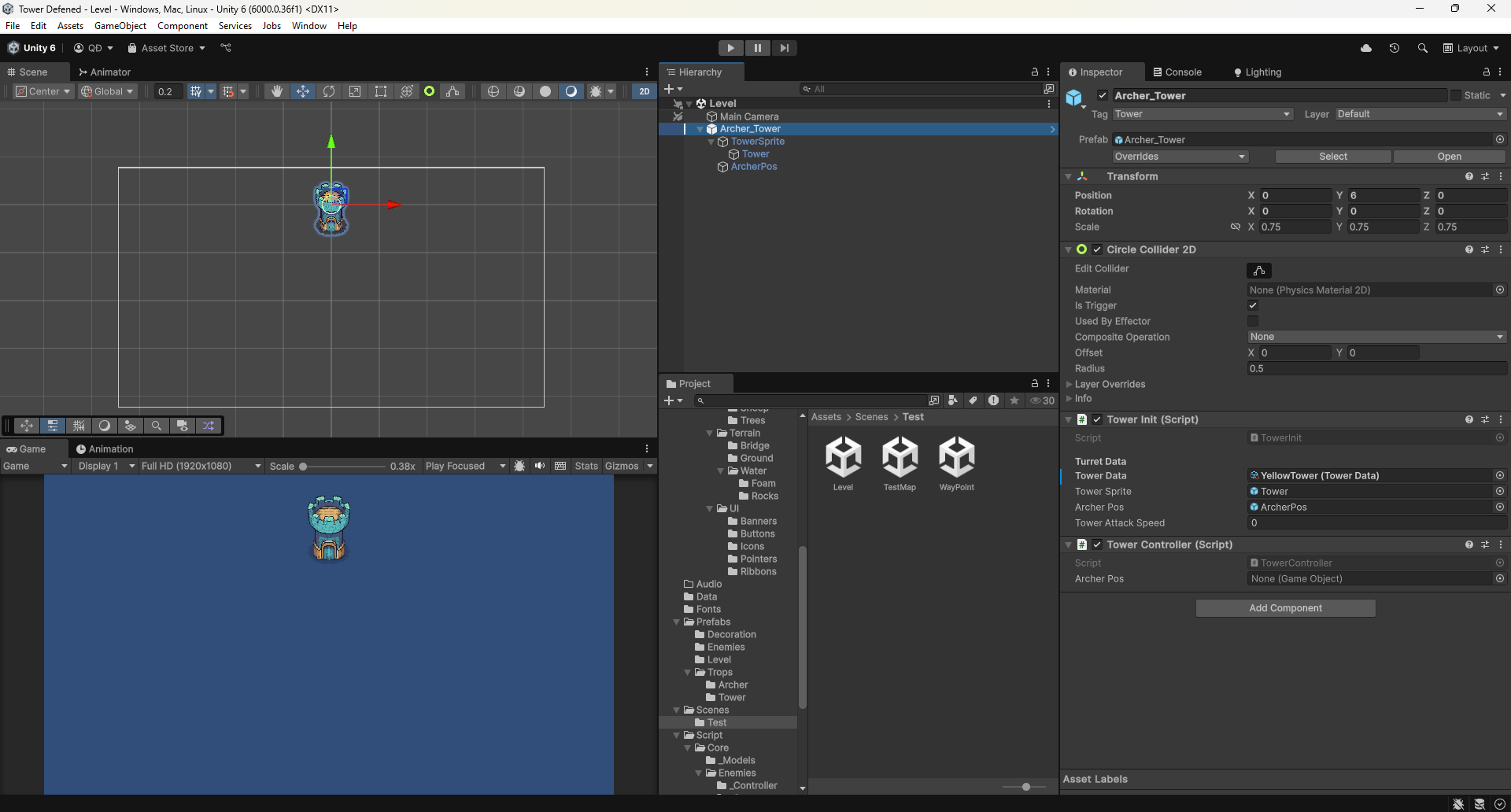


*Hình 3.1.3.a Khởi tạo thuật toán gen map tự động*

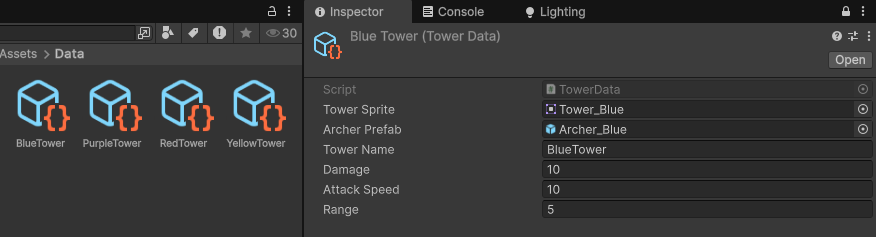
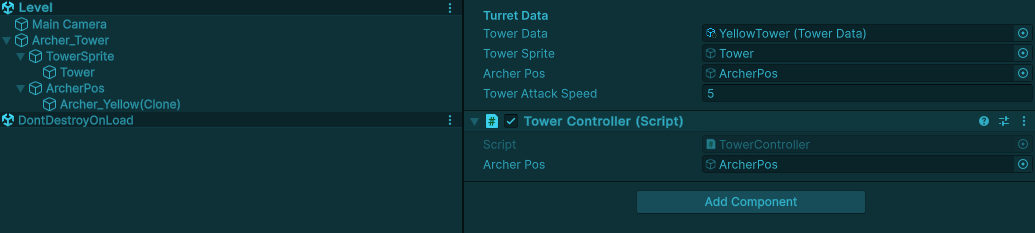
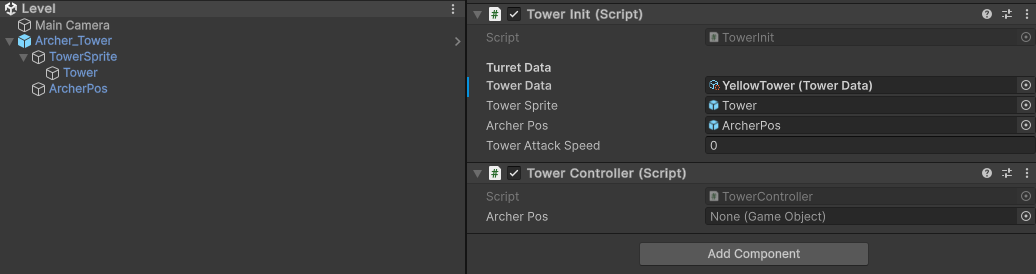
**

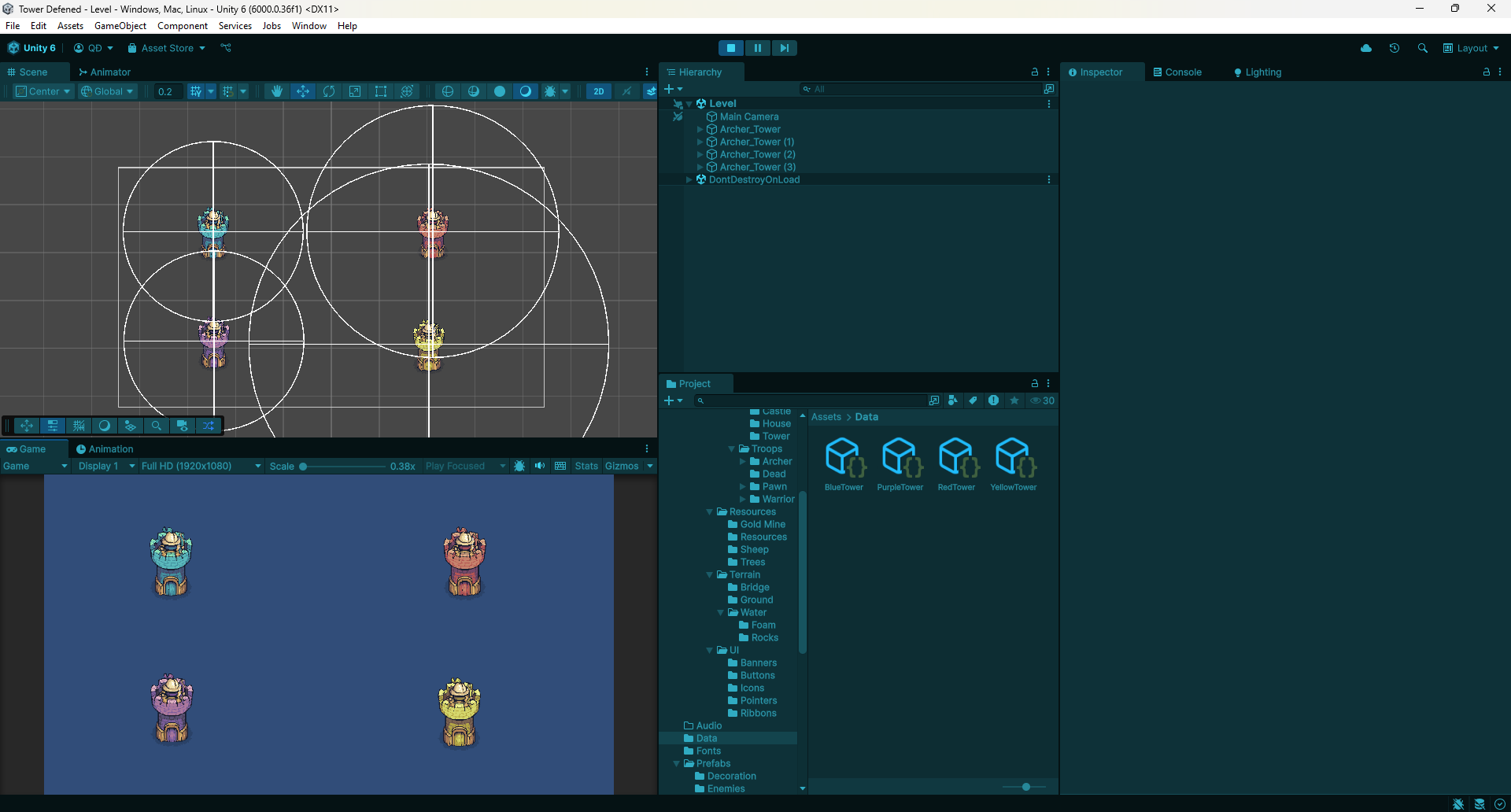
*Hình 3.1.3.b Thêm vật trang trí tự động*

1. **Tower System**

****

Hình 3.1.4.a Khởi tạo Tower

  
Hình 3.1.4.b Khởi tạo data cho tower  
  


  
Hình 3.1.4.c Tạo Tower theo data

1. **Kết hợp**



*Hình 3.1.5.a Kết hợp các thử nghiệm lại với nhau*

1. **Main menu**





1. **Game Menu**



*Hình 3.1.7.a Giao diện game*



*Hình 3.1.7.b Game*

# KẾT LUẬN

Qua quá trình thực hiện khóa luận này, đã đạt được những kết quả sau:

* Xây dựng được hệ thống lưu trữ dữ liệu linh hoạt cho người chơi thông qua cơ chế quản lý Save Slot.
* Tích hợp thành công thuật toán Quadtree nhằm tối ưu hóa xử lý không gian trong game, giúp cải thiện hiệu suất khi quản lý các đối tượng trong bản đồ.
* Áp dụng kỹ thuật sinh bản đồ ngẫu nhiên (Procedural Map Generation) để tạo ra trải nghiệm chơi phong phú, không lặp lại giữa các lần chơi.
* Giao diện người dùng được thiết kế trực quan, giúp người chơi dễ dàng tương tác với các chức năng như lựa chọn cấp độ, lưu game, và bắt đầu trò chơi.

Ưu điểm:

* Tối ưu hiệu suất nhờ vào thuật toán Quadtree trong việc phân vùng và truy xuất dữ liệu không gian.
* Tăng tính hấp dẫn và khả năng tái chơi nhờ vào bản đồ sinh ngẫu nhiên.
* Quy trình lưu và tải dữ liệu được xử lý rõ ràng, có kiểm tra và phản hồi theo trạng thái dữ liệu.
* Cấu trúc mã và luồng chương trình rõ ràng, dễ mở rộng và bảo trì.

**Điểm hạn chế:**

* Hệ thống bản đồ ngẫu nhiên hiện tại còn đơn giản, chưa có sự điều chỉnh linh hoạt về độ khó hoặc môi trường chơi.
* Chưa áp dụng các kỹ thuật nén hoặc tối ưu dữ liệu lưu trữ để tiết kiệm bộ nhớ.
* Hệ thống lưu game mới dừng ở mức cơ bản, chưa hỗ trợ nhiều loại dữ liệu phức tạp như trạng thái AI, nhiệm vụ,...

**Hướng phát triển**

* Phát triển thêm hệ thống lưu nâng cao hỗ trợ nhiều yếu tố gameplay hơn (AI, vật phẩm, sự kiện,...).
* Mở rộng thuật toán bản đồ ngẫu nhiên với các quy tắc sinh phức tạp hơn như Perlin Noise, Cellular Automata,...

# TÀI LIỆU THAM KHẢO

**Tài liệu tham khảo:**

[**Unity Discussions - A Space to Discuss All Things Unity**](https://discussions.unity.com/)

[**Garnet's Tutorials - YouTube**](https://www.youtube.com/playlist?list=PLsPeQQshJs0z6sNzZz30da9Sc5Vx7dyEs)

[**Procedural map generation using chess maze - YouTube**](https://www.youtube.com/playlist?list=PLcRSafycjWFfQJMXmmHIc9pE_egT21aCX)

[**Data Structures: Making a QuadTree in Unity - YouTube**](https://www.youtube.com/watch?v=7PWRYjgqoUg)

[**Jobs System, QuadTree, Morton Code - Siêu tối ưu thuật toán Boid bằng đa luồng và cây tứ phân - YouTube**](https://www.youtube.com/watch?v=7rXd-c81YJA&t=1563s)