

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP
KHOA ĐIỆN TỬ



ĐỒ ÁN TỐT NGHIỆP

BỘ MÔN : TIN HỌC CÔNG NGHIỆP

**ĐỀ TÀI: PHÁT TRIỂN TRÒ CHƠI DUNGEON AND GUN
BẰNG PHẦN MỀM UNITY**

GIÁO VIÊN HƯỚNG DẪN : TRẦN THỊ THANH
HỌ VÀ TÊN SINH VIÊN : ĐẶNG HOÀNG NAM
LỚP : K55KMT
MSSV : K195480106015

THÁI NGUYÊN - 2024

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP
KHOA ĐIỆN TỬ



ĐỒ ÁN TỐT NGHIỆP

BỘ MÔN : TIN HỌC CÔNG NGHIỆP

**ĐỀ TÀI: PHÁT TRIỂN TRÒ CHƠI DUNGEON AND GUN
BẰNG PHẦN MỀM UNITY**

GIÁO VIÊN HƯỚNG DẪN : TRẦN THỊ THANH
HỌ VÀ TÊN SINH VIÊN : ĐẶNG HOÀNG NAM
LỚP : K55KMT
MSSV : K195480106015

THÁI NGUYÊN - 2024

TRƯỜNG ĐHKTCN

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

KHOA ĐIỆN TỬ

Độc lập - Tự do - Hạnh phúc

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

Sinh viên: Đặng Hoàng Nam

Lớp: K55KMT

Khoá: 2019-2024

Bộ môn: Tin học công nghiệp

Giáo viên hướng dẫn: Trần Thị Thanh

1. Tên đề tài tốt nghiệp

Phát triển trò chơi Dungeon and Gun bằng phần mềm Unity

2. Các số liệu ban đầu (nếu có)

3. Nội dung các phần thuyết minh và tính toán

- Khảo sát bài toán.*
- Giới thiệu về phần mềm Unity và ngôn ngữ lập trình.*
- Phân tích thiết kế hệ thống*
- Phân tích, lập trình và kiểm thử trò chơi*
- Tổng kết những kết quả đạt được và hướng phát triển của đồ án*

5. Ngày giao nhiệm vụ : 06/12/2023

6. Ngày hoàn thành nhiệm vụ : 08/03/2024

BCN KHOA

TRƯỞNG BỘ MÔN

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

(Ký và ghi rõ họ tên)

(Ký và ghi rõ họ tên)

TRƯỜNG ĐHKTCN
KHOA ĐIỆN TỬ

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

PHIẾU GHI ĐIỂM
HƯỚNG DẪN ĐỒ ÁN TỐT NGHIỆP

Sinh viên: Đặng Hoàng Nam

Lớp: K55KMT

GVHD: Trần Thị Thanh

Đề tài: Phát triển trò chơi Dungeon and Gun bằng phần mềm Unity

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....
.....
.....
.....

Xếp loại: Điểm :

Thái Nguyên, ngày....tháng.....năm 20....

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

MỤC LỤC

LỜI CẢM ƠN.....	1
LỜI MỞ ĐẦU.....	2
CHƯƠNG I. TỔNG QUAN VỀ ĐỀ TÀI.....	3
1.1. Giới thiệu về trò chơi.....	3
1.2. Mô tả trò chơi.....	4
1.3. Yêu cầu của đề tài.....	4
CHƯƠNG II. NGÔN NGỮ LẬP TRÌNH.....	5
VÀ MÔI TRƯỜNG LÀM VIỆC	5
2.1. Unity Engine.....	5
2.1.1. Unity Engine là gì ?.....	5
2.1.2. Các tính năng cơ bản của Unity	6
2.1.3. Ưu điểm và nhược điểm của Unity	6
2.1.4. Các thành phần cơ bản trong Unity	8
2.2. Microsoft Visual Studio.....	13
2.2.1. Visual Studio là gì ?.....	13
2.2.2. Một số tính năng của Visual Studio	14
CHƯƠNG III. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.....	16
3.1. Các chức năng của hệ thống	16
3.2. Biểu đồ Use-Case.....	16
3.3. Biểu đồ hoạt động	17
3.4. Biểu đồ tuần tự.....	18
3.5. Biểu đồ trạng thái	19
3.6. Biểu đồ thành phần.....	19
CHƯƠNG IV. LẬP TRÌNH VÀ KIỂM THỬ.....	20
4.1. Giao diện người dùng	20

4.1.1. Màn hình menu chính	20
4.1.2. Hướng dẫn trò chơi	20
4.1.3. Các màn chơi chính.....	21
4.2. Thiết kế trò chơi.....	23
4.2.1. Canvas	23
4.2.2. Background	24
4.2.3. Player	26
4.2.4. Kẻ địch	28
4.3. Xây dựng trò chơi trên nền tảng máy tính.....	31
CHƯƠNG V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	35
5.1. Nhận xét và kết luận	35
5.2. Hướng phát triển của đề án.....	35
DANH MỤC HÌNH ẢNH.....	36
TÀI LIỆU THAM KHẢO	38

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc đối với cô **Trần Thị Thanh**, giảng viên của trường Đại Học Kỹ Thuật Công Nghiệp Thái Nguyên đã nhiệt tình hướng dẫn hướng dẫn em hoàn thành tốt báo cáo này.

Em xin gửi lời cảm ơn chân thành tới các thầy, các cô là giảng viên khoa điện tử cùng các thầy, cô bộ môn tin học công nghiệp trường Đại Học Kỹ Thuật Công Nghiệp Thái Nguyên đã tận tình truyền đạt kiến thức và hướng dẫn cho em trong suốt quá trình học tập tại trường.

Trong quá trình làm báo cáo khó tránh khỏi sai sót, rất mong các thầy, cô bỏ qua, em rất mong nhận được ý kiến góp ý của thầy cô bộ môn để em rút kinh nghiệm cho tương lai.

Em xin chân thành cảm ơn!

LỜI MỞ ĐẦU

Trong thời đại công nghệ thông tin như hiện nay, sản phẩm công nghệ ngày càng chịu sự đánh giá khắt khe từ phía người dùng, trong đó có các sản phẩm trò chơi điện tử. Ở nước ta hiện nay, việc xây dựng một ứng dụng trò chơi đã không còn xa lạ, nhưng để tạo ra một trò chơi hay và chất lượng là một vấn đề không hề dễ dàng. Ngoài việc người lập trình cần có kinh nghiệm, kiến thức vững chắc về các công cụ lập trình thì cần phải hiểu, hoạch định và phân tích tối ưu được các vai trò chức năng của hệ thống thông tin.

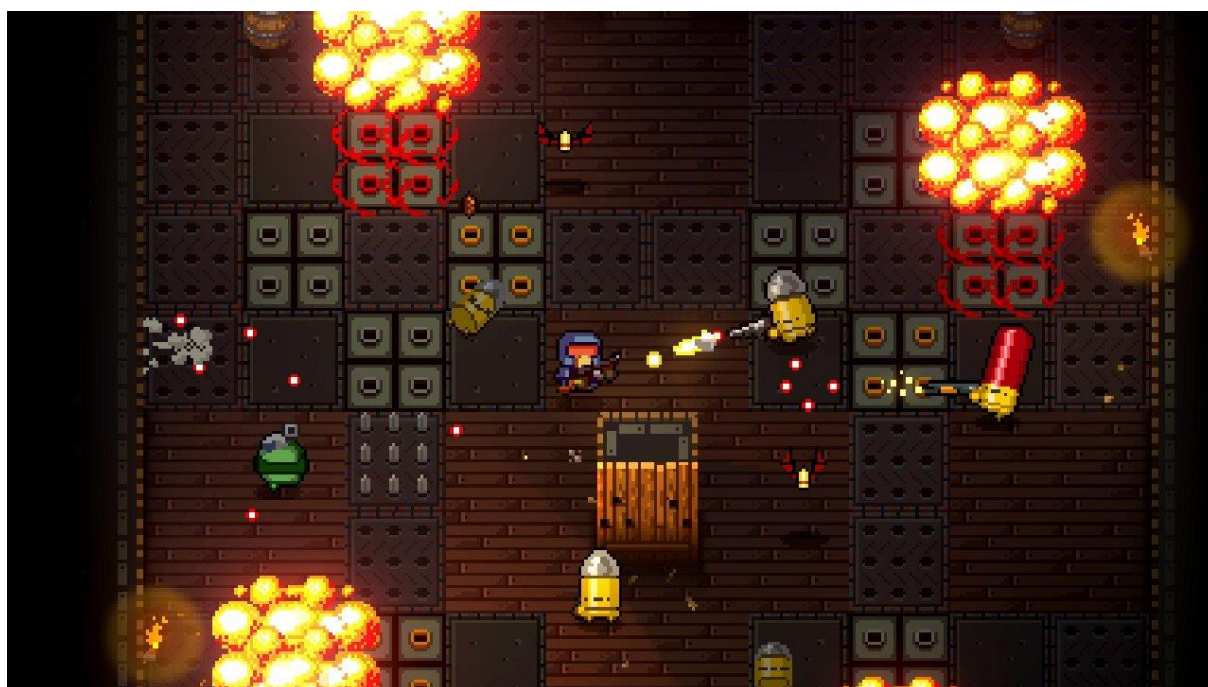
Ngành công nghiệp game hiện nay có thể nói là bùng nổ với tốc độ phát triển chóng mặt, rất nhiều những trò chơi hay và hấp dẫn được ra đời trong thời gian qua. Phía sau sự thành công của đó là công sức và sự sáng tạo của các nhà phát triển, đi cùng với một công cụ hỗ trợ lập trình giúp việc thiết kế trực quan và nhanh chóng, tiết kiệm thời gian sản xuất.

Từ xu hướng phát triển trên, đề án này sẽ khảo sát và nghiên cứu và Unity Engine – Một Game Engine rất phổ biến và không kém phần mạnh mẽ hiện nay nhằm thực nghiệm việc phát triển trò chơi bắn súng 2D có tên **“Dungeon and Gun”**. Với đề án này, em mong muốn có thể áp dụng những kiến thức đã học ở trường vào thực tế, bên cạnh đó em hy vọng rằng đề án này sẽ là một sản phẩm chất lượng, mang đến cho người chơi trải nghiệm thú vị và sẽ là một tài liệu tham khảo có ích cho những bạn có chung niềm đam mê với lập trình game.

CHƯƠNG I. TỔNG QUAN VỀ ĐỀ TÀI

1.1. Giới thiệu về trò chơi

Dungeon and Gun - tựa Tiếng Việt là “**Hầm ngục và Súng**” là trò chơi hành động và bắn súng được thiết kế trong môi trường 2D. Trò chơi này có góc nhìn từ trên xuống dưới, người chơi sẽ điều khiển nhân vật di chuyển, tránh né và sử dụng súng để tiêu diệt kẻ địch. Kẻ địch đa dạng về hình dáng, cách thức di chuyển và tấn công, khiến trò chơi trở nên thú vị và thử thách.



Hình 1.1. Trò chơi hành động bắn súng 2D

Trò chơi được thiết kế với phong cách khối vuông đơn giản – còn gọi là phong cách Pixel Art, mang lại cảm giác hoài cổ về những trò chơi cổ điển ngày xưa. Nhờ sự đơn giản của mình, trò chơi có chi phí phát triển rất thấp, phù hợp với những nhà phát triển độc lập hoặc công ty quy mô nhỏ. Điều này giúp những nhà phát triển có thể đưa những ý tưởng sáng tạo của mình tiếp cận đến thị trường mà không cần đầu tư quá nhiều vào công đoạn thiết kế hình ảnh trò chơi.

1.2. Mô tả trò chơi

Người chơi sẽ nhập vai vào một anh chàng với một khẩu súng phóng tên lửa trên tay. Nhiệm vụ của người chơi là đi qua các hầm ngục và đối đầu với các kẻ thù nguy hiểm. Mỗi hầm ngục sẽ có một kẻ thù với cách di chuyển, tấn công khác nhau và độ khó của chúng cũng sẽ tăng dần.

Sau khi vào trò chơi, người chơi được đưa đến một hầm ngục khởi đầu. Ở đây người chơi sẽ học cách thao tác cơ bản của game như di chuyển, lướt né tránh, tấn công. Sau khi hoàn thành màn hướng dẫn, người chơi đi qua cổng dịch chuyển để đến hầm ngục chính. Tại đây người chơi sẽ lần lượt vượt qua các hầm ngục với nhiều loại kẻ thù khác nhau. Khi bị dính đòn, người chơi sẽ mất máu. Khi lượng máu này trở về 0, người chơi sẽ chết hay GameOver. Khi thua sẽ có màn hình GameOver hiện ra để người chơi chọn tiếp tục chơi hoặc về màn hình chính. Khi người chơi tiêu diệt kẻ thù ở một màn, sẽ có một cổng dịch chuyển hiện lên dẫn đến màn tiếp theo.

1.3. Yêu cầu của đề tài

- ✓ Tìm hiểu và nắm vững các thành phần trong Unity Engine để áp dụng vào xây dựng trò chơi
- ✓ Trò chơi được tối ưu tốt, tốc độ xử lý nhanh, không bị lỗi trong quá trình chơi
- ✓ Giao diện trò chơi dễ nhìn, thân thiện với người sử dụng, có tiếng Việt
- ✓ Công việc tính toán (va chạm, tấn công, hiệu ứng,...) phải được thực hiện chính xác, không có sai sót

CHƯƠNG II. NGÔN NGỮ LẬP TRÌNH VÀ MÔI TRƯỜNG LÀM VIỆC

2.1. Unity Engine

2.1.1. Unity Engine là gì ?

Unity Engine là một môi trường phát triển trò chơi và ứng dụng đa nền tảng mạnh mẽ được phát triển bởi công ty Unity Technologies. Nó cho phép nhà phát triển tạo ra các trò chơi 2D, 3D, ứng dụng tương tác thực tế ảo (VR) và thực tế tăng cường (AR), cũng như các ứng dụng đa nền tảng khác. Unity Engine là một trong những công cụ phát triển phổ biến và mạnh mẽ trong ngành công nghiệp trò chơi và phát triển ứng dụng hiện nay.



Hình 2.1. Unity Engine

Unity mang đến sức mạnh kỳ diệu cho nhân vật mà chúng ta muốn thể hiện sống động hơn trong không gian ba chiều đầy huyền ảo. Công nghệ cao này tạo ra một bước đột phá mới về sự khác biệt trong công nghệ làm game hiện nay, mang đến cho người chơi 1 cảm giác rất khác lạ và hào hứng trong từng chuyển động, tương lai công nghệ này được áp dụng vào game Việt Nam sẽ mở ra một trang mới trong thế giới game 2D, 3D huyền ảo.

2.1.2. Các tính năng cơ bản của Unity

- Tạo giao diện UI của Game như tạo drop bar, textbox,...
- Hiển thị mô hình 3D, 2D – hệ thống vật lý 2D, 3D
- Networking: Hỗ trợ tạo game chơi nhiều người cùng lúc
- Hỗ trợ nền tảng đặc biệt mới: AR – Augmented reality (Thực tế tăng cường), VR – Virtual reality (Thực tế ảo)
- AI – Hỗ trợ con bot trong màn hình game, hỗ trợ package tạo nên bot trong game
- Hỗ trợ hiển thị chữ bằng font đặc biệt

2.1.3. Ưu điểm và nhược điểm của Unity

Ưu điểm:

- **Editor:** Với Editor, nhà phát triển không cần thiết phải viết Code để sắp đặt các đối tượng trong Game như những Engine khác mà Developer có thể kéo thả, thay đổi vị trí của từng đối tượng trong Game trực tiếp trên Editor.
- **Đa nền tảng:** Việc bạn tạo ra Game mà Game đó có thể chạy được trên hầu hết những hệ điều hành quan trọng như Desktop (Mac, Window và Linux), Mobile (iOS, Android) hoặc Web thì cũng đã tiết kiệm công sức cũng như chi phí rất nhiều cho doanh nghiệp.

- **Miễn phí:** Và yếu tố cuối cùng chính là chi phí. Với Unity, miễn phí là một điểm thu hút rất nhiều Developer chọn làm việc với game engine này. Tuy nhiên, với các game được tạo ra miễn phí thì bắt buộc phải có Logo Unity trong Game.

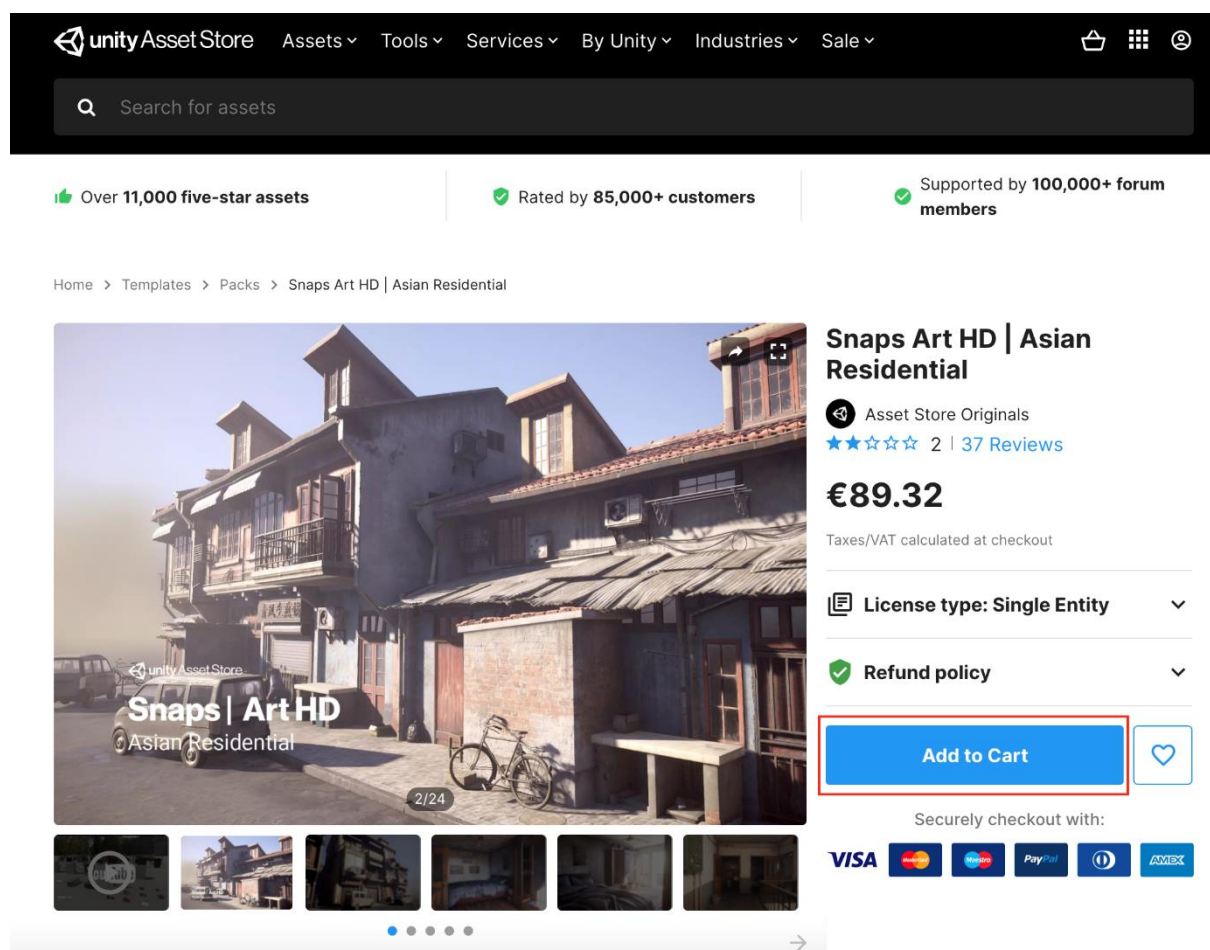
Nhược điểm:

- **Dung lượng Unity game bundle khá lớn:** So với những game engine khác, Unity sản xuất game có dung lượng nặng nên đây là một điểm trừ lớn. Thậm chí, game web do Unity sản xuất có thể có dung lượng lên đến cả trăm MB nên web chạy không nổi. Chính vì thế, cũng cùng một game đó thì game mobile lại chạy tốt trong khi game web lại giật, lag, không ổn định.
- **Các phiên bản cập nhật:** Unity cho ra mắt nhiều phiên bản cập nhật liên tục trong một năm và nhiều năm liên tục. Nếu phiên bản cập nhật được ra mắt trong cùng một năm thì vẫn có thể sử dụng song song nhiều phiên bản cùng năm. Còn nếu trong trường hợp bản cập nhật khác năm, và trong nội bộ một team, hoặc giữa team phát triển và khách hàng, sử dụng các phiên bản Unity khác năm với nhau thì khi push code, đẩy code lên cho member thì sẽ bị lỗi hình ảnh, script, code,... thậm chí là mất hết tiến độ. Tuy nhiên, điều khó là nếu như bạn nhận nhiều dự án cùng một lúc mà mỗi khách hàng lại sử dụng một phiên bản Unity khác nhau để phát triển game thì bạn bắt buộc phải thay đổi qua lại.

2.1.4. Các thành phần cơ bản trong Unity

❖ Assets

Assets là những tài nguyên được sử dụng trong việc phát triển game trong Unity. Các Assets này có thể là hình ảnh, mô hình 3D, âm thanh, hiệu ứng, ... được tạo ra bởi các nhà phát triển, có thể được download miễn phí hoặc trả phí. Tính năng này giúp giảm thiểu rất nhiều thời gian cho việc thiết kế và lập trình game. Các Assets được đăng tải trên trang chính thức [Unity Asset Store](https://unity.com/asset-store).

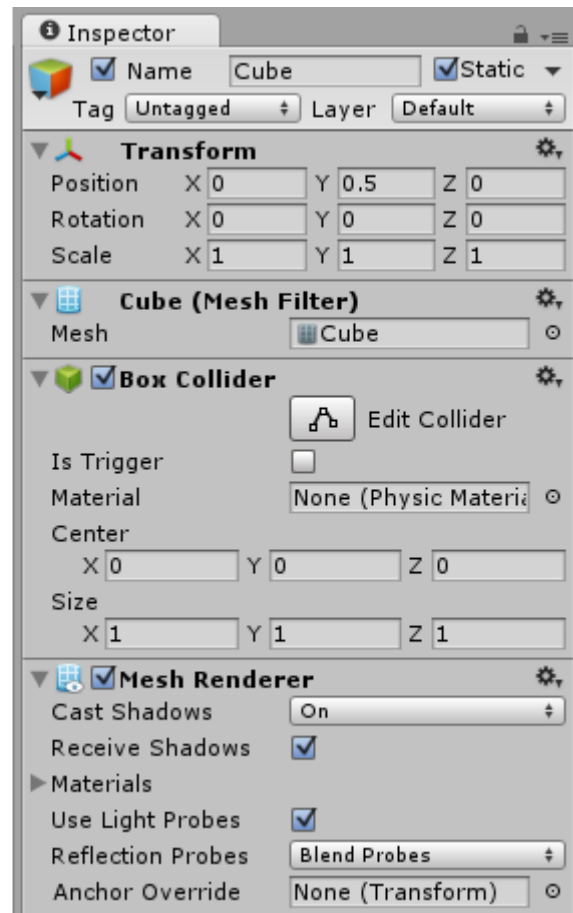
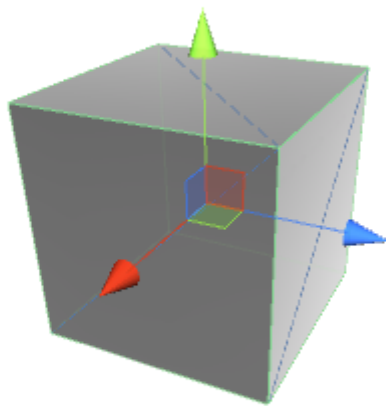


Hình 2.2. Unity Assets Store

Unity có một cộng đồng phát triển đông đảo. Bạn có thể tìm thấy hàng ngàn tài nguyên hình ảnh, âm thanh, công cụ hỗ trợ,... thậm chí là tài liệu hướng dẫn, tài liệu trực tuyến và các nguồn hỗ trợ từ cộng đồng để giúp bạn giải quyết các vấn đề phát triển.

❖ GameObject

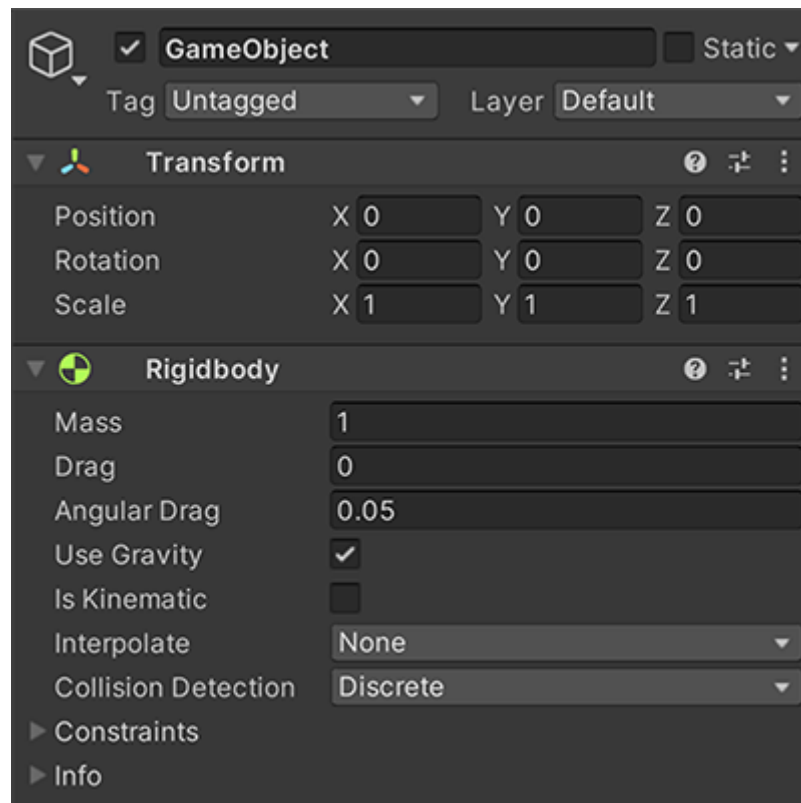
GameObject là đối tượng đại diện cho các Assets trong game như nhân vật, cây cỏ, công cụ, đạo cụ, camera, hiệu ứng, ... Các GameObject đều chứa một thuộc tính cơ bản là Transform, dùng để thiết lập vị trí, hướng và kích thước của GameObject.



Hình 2.3. GameObject

❖ Components

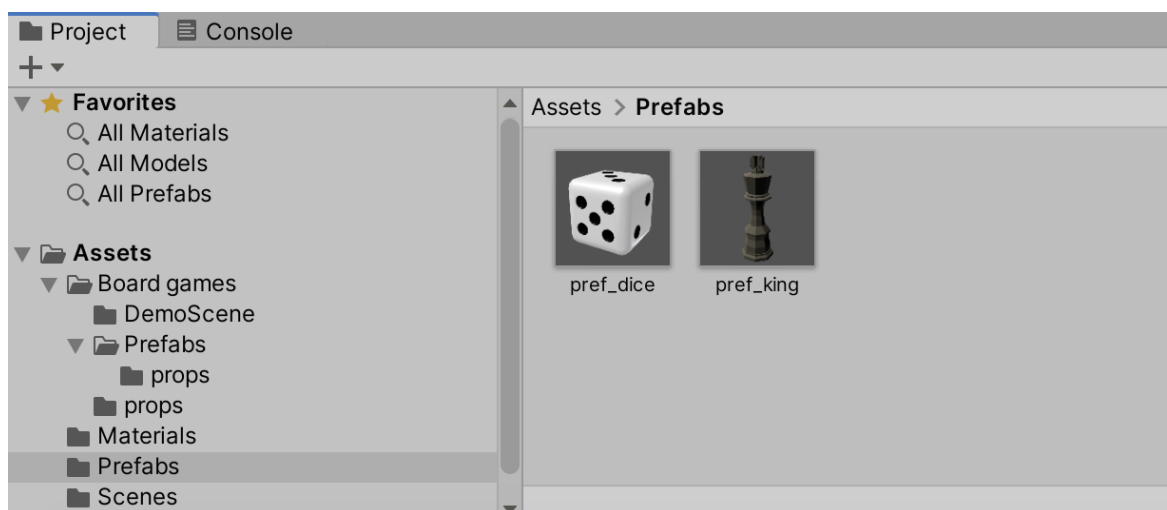
Components là các thuộc tính thêm vào GameObject như là Animation, âm thanh, mô hình 3D, hiệu ứng, ... nhằm xây dựng, kết hợp các yếu tố khác nhau để xác định các hình thái, hành vi, hiệu ứng, ... của đối tượng mong muốn trong game.



Hình 2.4. Components

❖ Prefabs

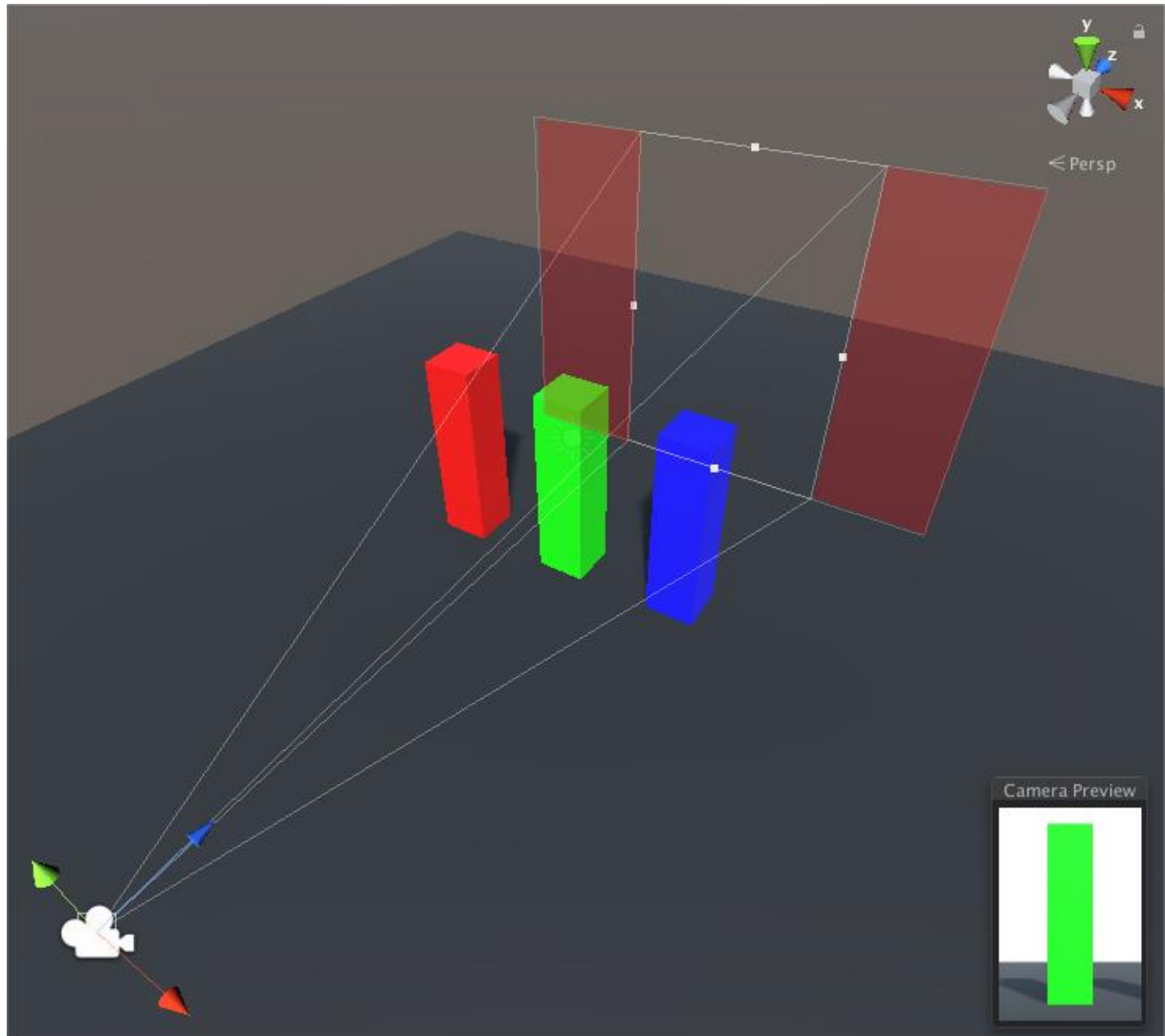
Prefabs là một GameObject hoàn chỉnh sau khi thêm các thuộc tính và được lưu trữ lại để tái sử dụng. Các GameObject được nhân bản từ một prefab sẽ giống nhau hoàn toàn. Để tạo ra một prefab, ta chỉ cần kéo một GameObject từ cửa sổ Hierarchy vào cửa sổ Project.



Hình 2.5. Prefabs

❖ Camera

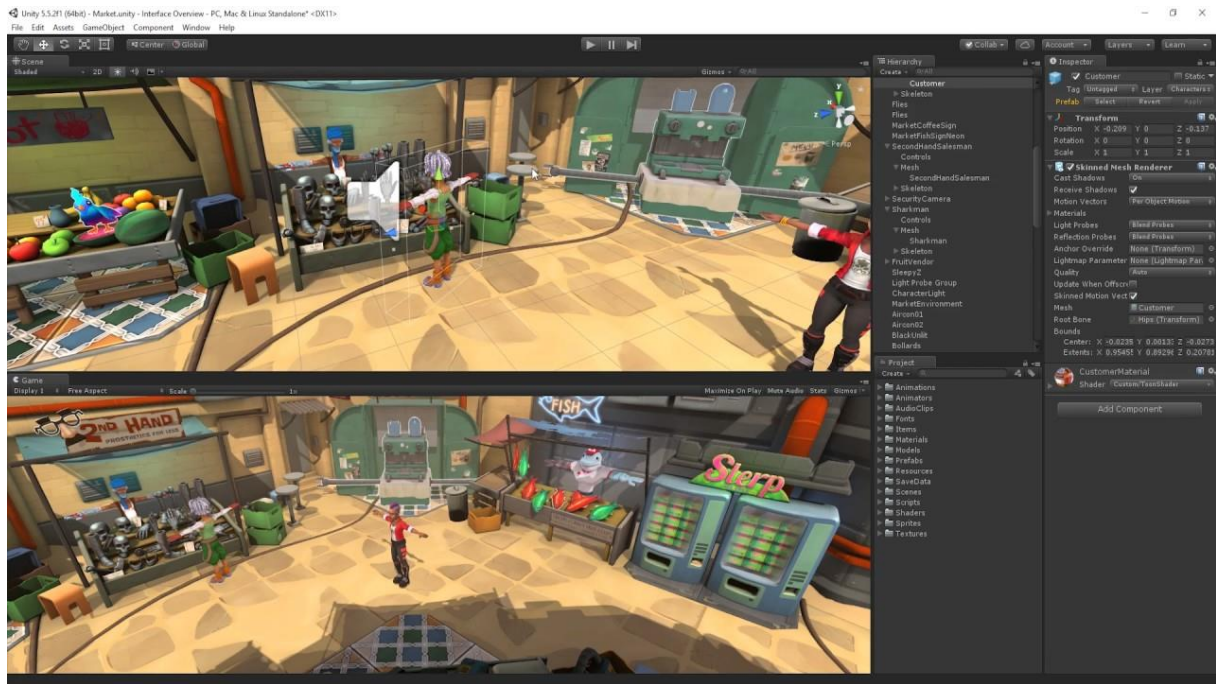
Camera trong Unity dùng để thể hiện khung hình, góc nhìn mà người chơi có thể nhìn thấy được trong game. Trong một game có thể thiết lập nhiều camera để chia nhỏ màn hình người chơi, tạo các hiệu ứng, hoặc tùy chỉnh như nhìn được từ phía sau, bản đồ thu nhỏ, ...



Hình 2.6. Camera

❖ Scenes

Trong Unity, Scene là một cảnh game, không gian game nơi thiết lập bố cục của các GameObjects, hoặc là một phần chứa các thiết lập giao diện như các menu trong game. Tạo ra nhiều Scenes sẽ giúp linh hoạt trong việc phân phối tối ưu tài nguyên, quản lý các phân đoạn trong game một cách độc lập.



Hình 2.7. Scenes

❖ Scripts

Scripts là một Component trong Unity. Đây là thành phần thiết yếu dùng để tương tác với các hành động của người chơi, hoặc quản lý các sự kiện để thay đổi chiều hướng của game tương ứng với kịch bản game. Unity cung cấp cho lập trình viên khả năng viết Script bằng ngôn ngữ C#.



Hình 2.8. Scripts

2.2. Microsoft Visual Studio

2.2.1. Visual Studio là gì ?

Microsoft Visual Studio là một môi trường phát triển tích hợp (IDE) từ Microsoft. Microsoft Visual Studio còn được gọi là "Trình soạn thảo mã nhiều người sử dụng nhất thế giới", được dùng để lập trình C++ và C# là chính. Nó được sử dụng để phát triển chương trình máy tính cho Microsoft Windows, cũng như các trang web, các ứng dụng web và các dịch vụ web. Nó có thể sản xuất cả hai ngôn ngữ máy và mã số quản lý.

Visual Studio bao gồm một trình soạn thảo mã hỗ trợ IntelliSense cũng như cải tiến mã nguồn. Trình gỡ lỗi tích hợp hoạt động cả về trình gỡ lỗi mức độ mã nguồn và gỡ lỗi mức độ máy. Công cụ tích hợp khác bao gồm một mẫu thiết kế các hình thức xây dựng giao diện ứng dụng, thiết kế web, thiết kế lớp và thiết kế giản đồ cơ sở dữ liệu. Nó chấp nhận các plug-in nâng cao các chức năng ở hầu hết các cấp bao gồm thêm hỗ trợ cho các hệ thống quản lý phiên bản (như Subversion) và bổ sung thêm bộ công cụ mới như biên tập và thiết kế trực quan cho các miền ngôn ngữ cụ thể hoặc bộ công cụ dành cho các khía cạnh khác trong quy trình phát triển phần mềm.



Hình 2.9. Microsoft Visual Studio

2.2.2. Một số tính năng của Visual Studio

- **Hỗ trợ nhiều ngôn ngữ lập trình:** Tính năng này giúp phát hiện bất kỳ lỗi hoặc tham chiếu ngôn ngữ chéo (cross-language reference) nào một cách dễ dàng.
- **Intelli-Sense:** Là một tính năng giúp phát hiện có bất kỳ đoạn code nào bị bỏ sót hay không, tự động thực thi cú pháp biến (variable syntaxes) và khai báo biến (variable declarations). Ví dụ: Nếu một biến nào đó đang được sử dụng trong chương trình và người dùng quên khai báo, intellisense sẽ khai báo biến đó cho người dùng.
- **Hỗ trợ đa nền tảng:** Visual Studio hoạt động trên cả 3 nền tảng Windows, Linux, Mac.
- **Tiện ích mở rộng và hỗ trợ:** Các extension hay tiện ích mở rộng của Visual Studio giúp tăng tốc quá trình phát triển ứng dụng mà không ảnh hưởng đến hiệu suất của editor.
- **Repository:** Visual Studio được kết nối với Git hoặc có thể được kết nối với bất kỳ Repository nào khác.
- **Code editor:** Visual Studio có class Code editor tốt nhất hỗ trợ nhiều chức năng. Nó cho phép bookmark trong code để kết hợp Quick Navigation. Visual Studio cũng có chức năng Incremental Search, Regex Search, Multi-item Clipboard và Task-list.
- **Web-Support:** Các ứng dụng web có thể được xây dựng và hỗ trợ trong Visual Studio.
- **Hỗ trợ Terminal:** Visual Studio hỗ trợ Terminal hoặc Console tích hợp giúp người dùng không cần chuyển đổi giữa hai màn hình.
- **Hỗ trợ Git:** Tài nguyên có thể được lấy từ Github Repo trực tuyến và ngược lại giúp tiết kiệm thời gian và công sức.

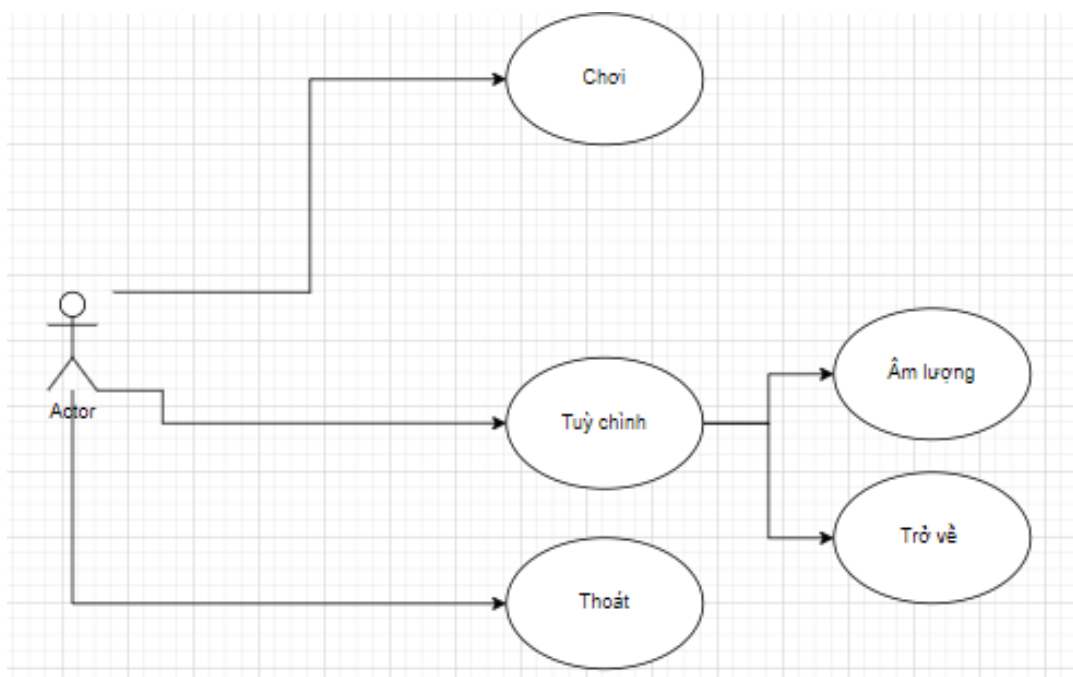
- **Debugger:** Đây là một tính năng hữu ích cho phép nhà phát triển kiểm tra trạng thái của chương trình và phát hiện bug ở đâu. Bạn cũng có thể xem source code chương trình của mình bằng cách sử dụng các công cụ gỡ lỗi của debugger.
- **Windows Forms Designer:** tạo các ứng dụng GUI bằng Windows Forms.
- **Class designer:** Class designer cho phép chỉnh sửa các class bao gồm các thành viên và quyền truy cập của chúng bằng cách sử dụng mô hình UML.
- **Web designer:** Visual Studio cũng hỗ trợ một trình soạn thảo và thiết kế trang web, cho phép tạo các trang web bằng cách kéo và thả các widget.
- **Mapping Designer:** Mapping Designer được LINQ to SQL sử dụng để thiết kế mapping giữa các lược đồ thông tin và từ đó các class sẽ đóng gói dữ liệu.
- **Data Designer:** Data Designer được sử dụng để chỉnh sửa các lược đồ thông tin, cũng như các bảng được viết, khóa chính và khóa ngoại và các ràng buộc (constraint).

CHƯƠNG III. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1. Các chức năng của hệ thống

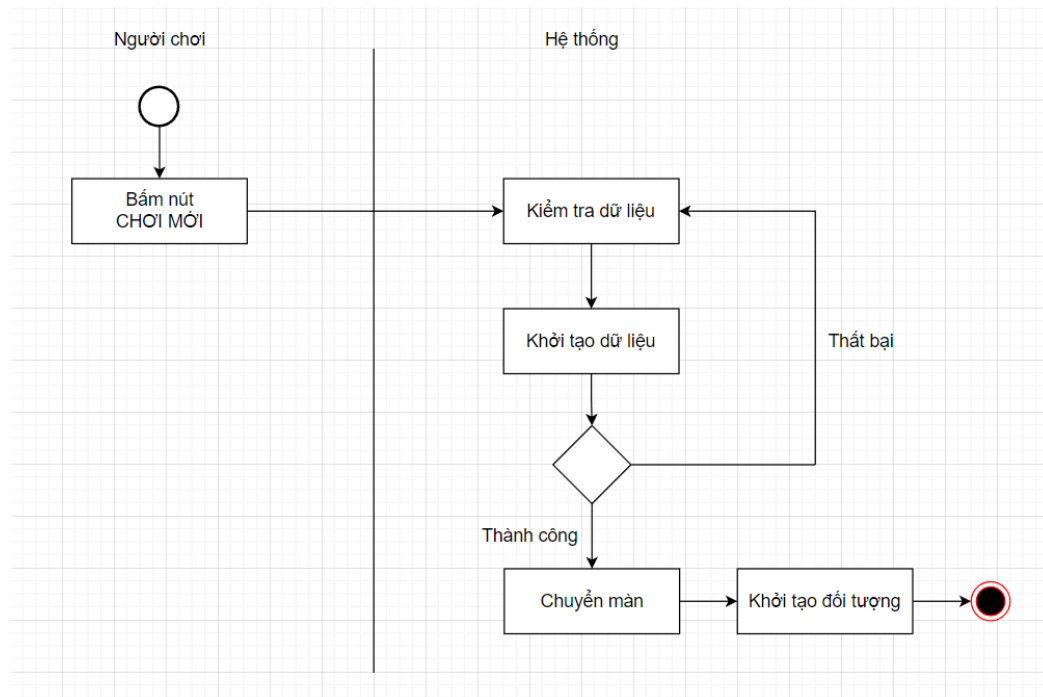
STT	Tên chức năng	Mô tả
1	Bắt đầu, cài đặt và thoát	Khi vào game sẽ có các tùy chọn như chơi mới, cài đặt và thoát
2	Điều khiển nhân vật	Di chuyển nhân vật theo các hướng trong môi trường 2D như trái, phải, lên, xuống
3	Tương tác với các đối tượng	Có thể tương tác với các đối tượng như công dịch chuyển, trụ bắn lửa, kẻ địch,...
4	Tấn công	Tấn công gây sát thương lên kẻ địch

3.2. Biểu đồ Use-Case

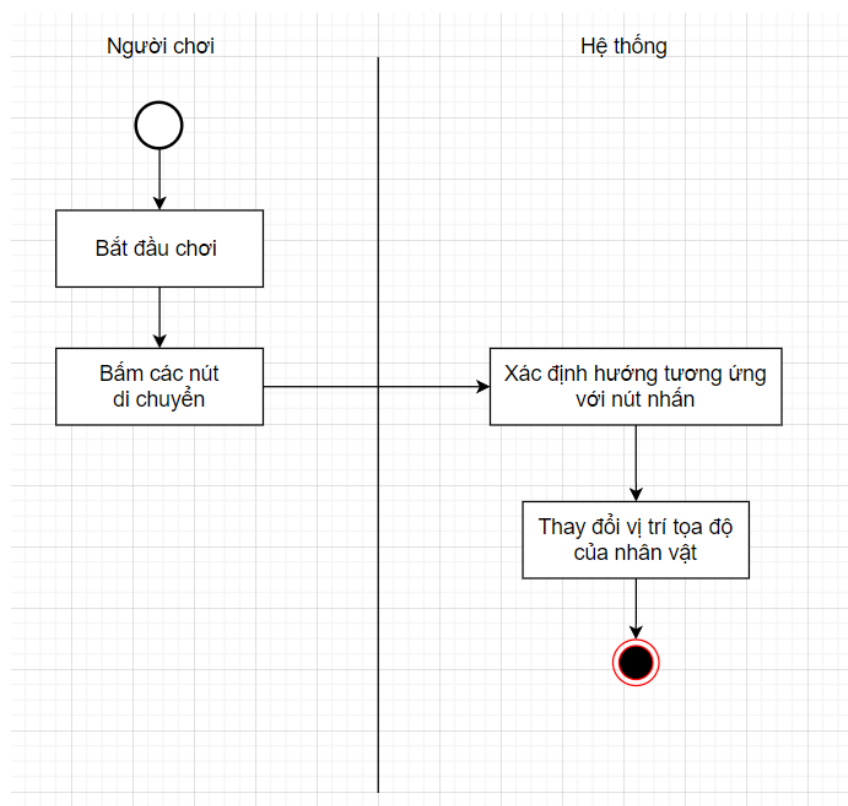


Hình 3.1. Biểu đồ Use-Case

3.3. Biểu đồ hoạt động

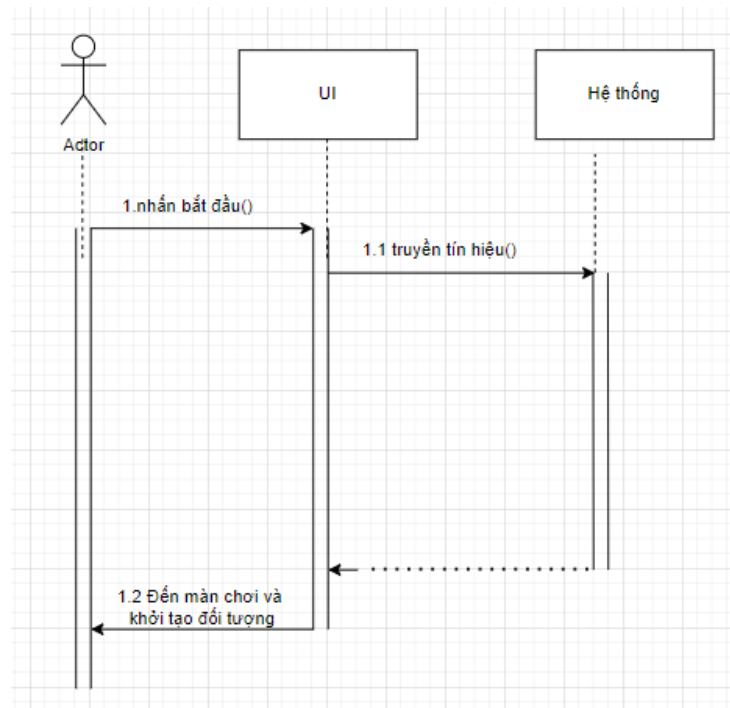


Hình 3.2. Biểu đồ hoạt động bắt đầu trò chơi

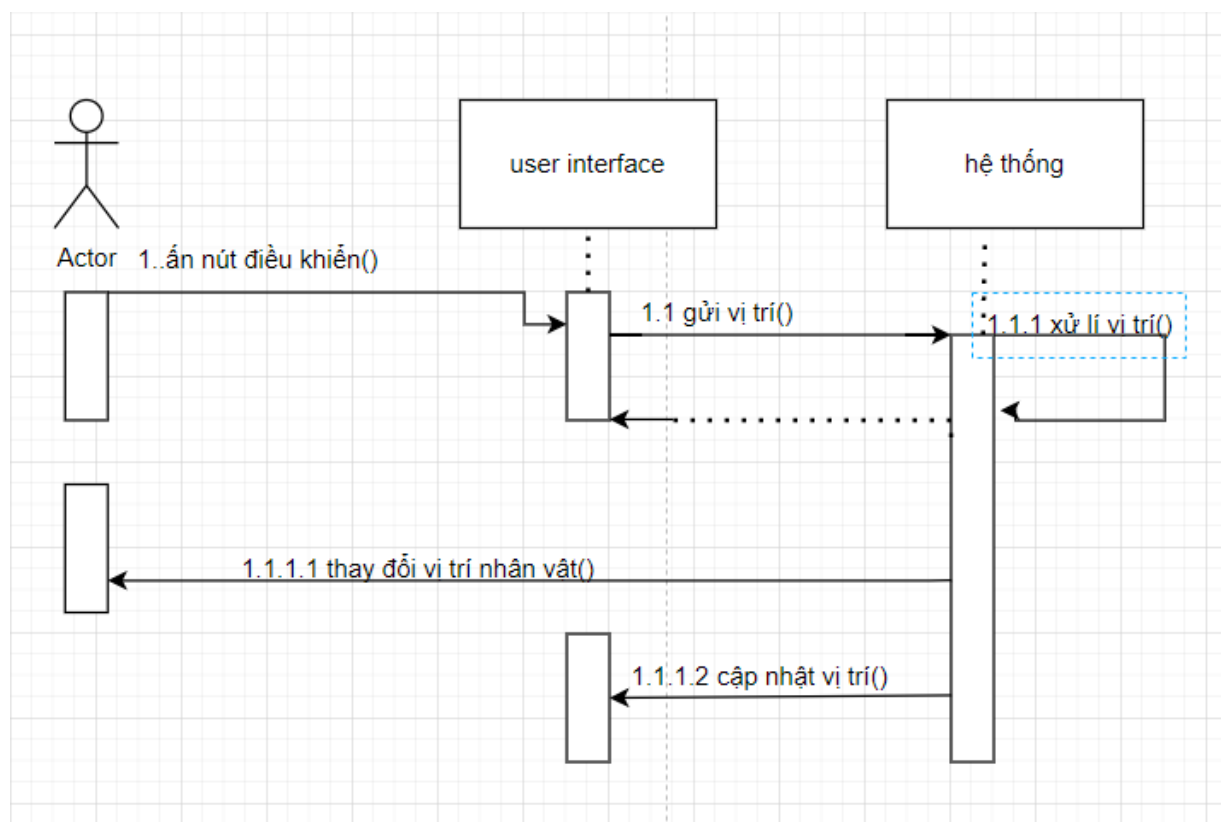


Hình 3.3. Biểu đồ điều khiển nhân vật

3.4. Biểu đồ tuần tự

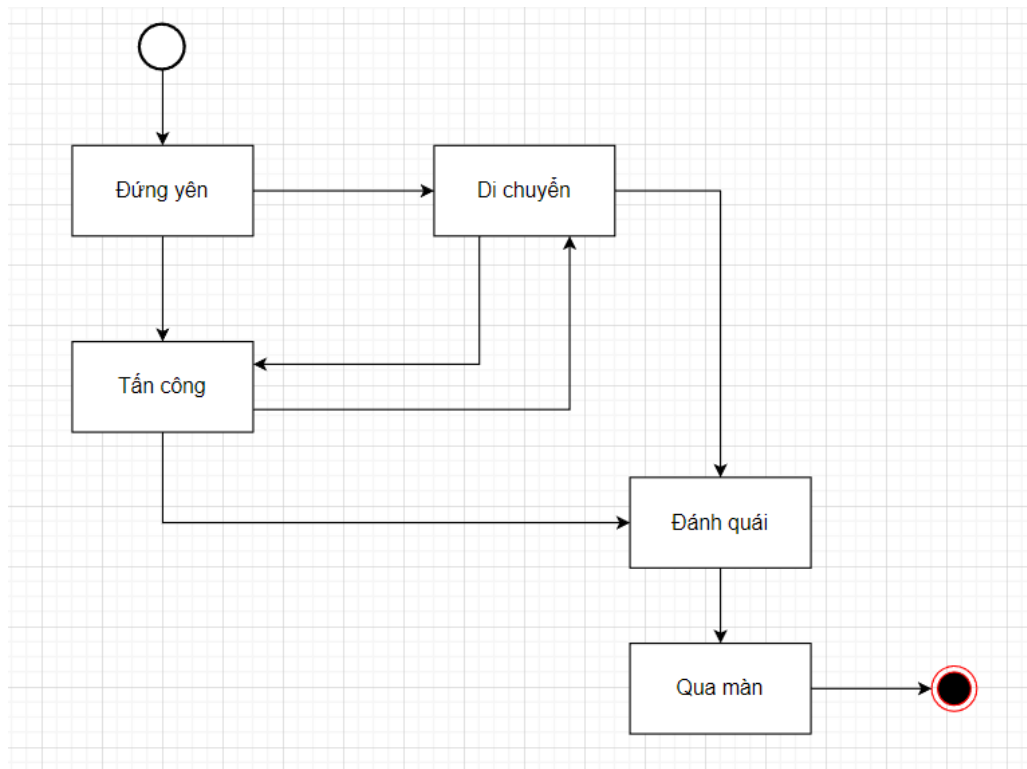


Hình 3.4. Biểu đồ tuần tự bắt đầu chơi



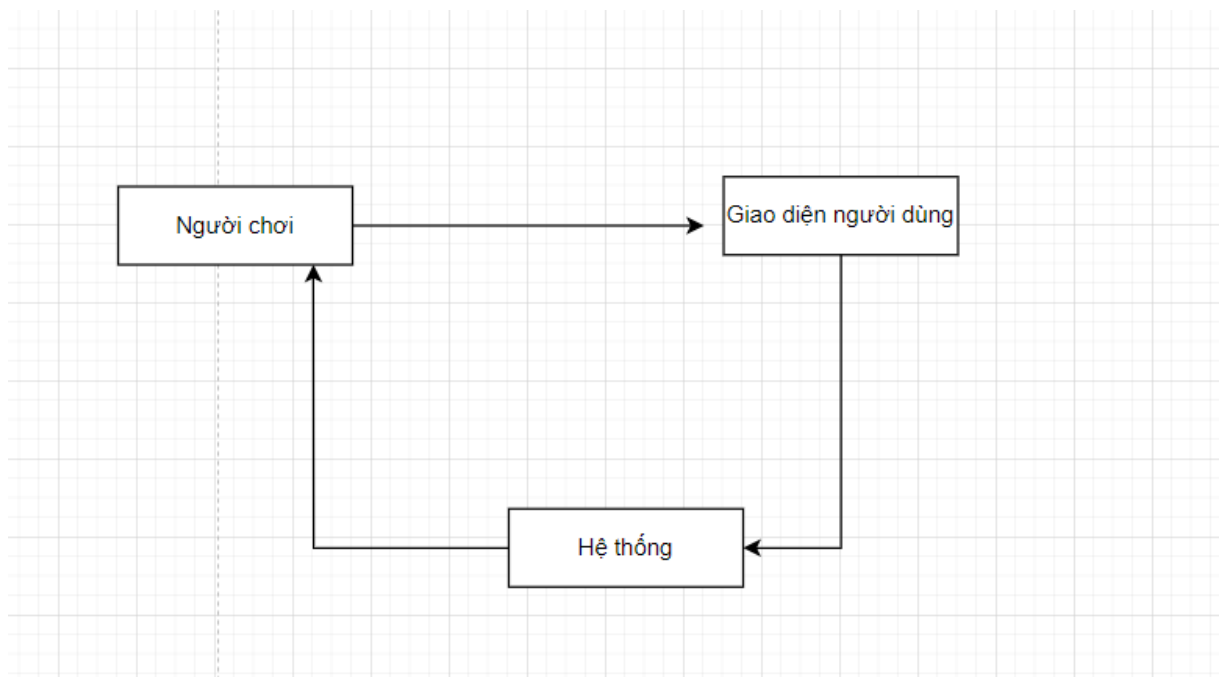
Hình 3.5. Biểu đồ tuần tự điều khiển nhân vật

3.5. Biểu đồ trạng thái



Hình 3.6. Biểu đồ trạng thái

3.6. Biểu đồ thành phần



Hình 3.7. Biểu đồ thành phần

CHƯƠNG IV. LẬP TRÌNH VÀ KIỂM THỬ

4.1. Giao diện người dùng

4.1.1. Màn hình menu chính

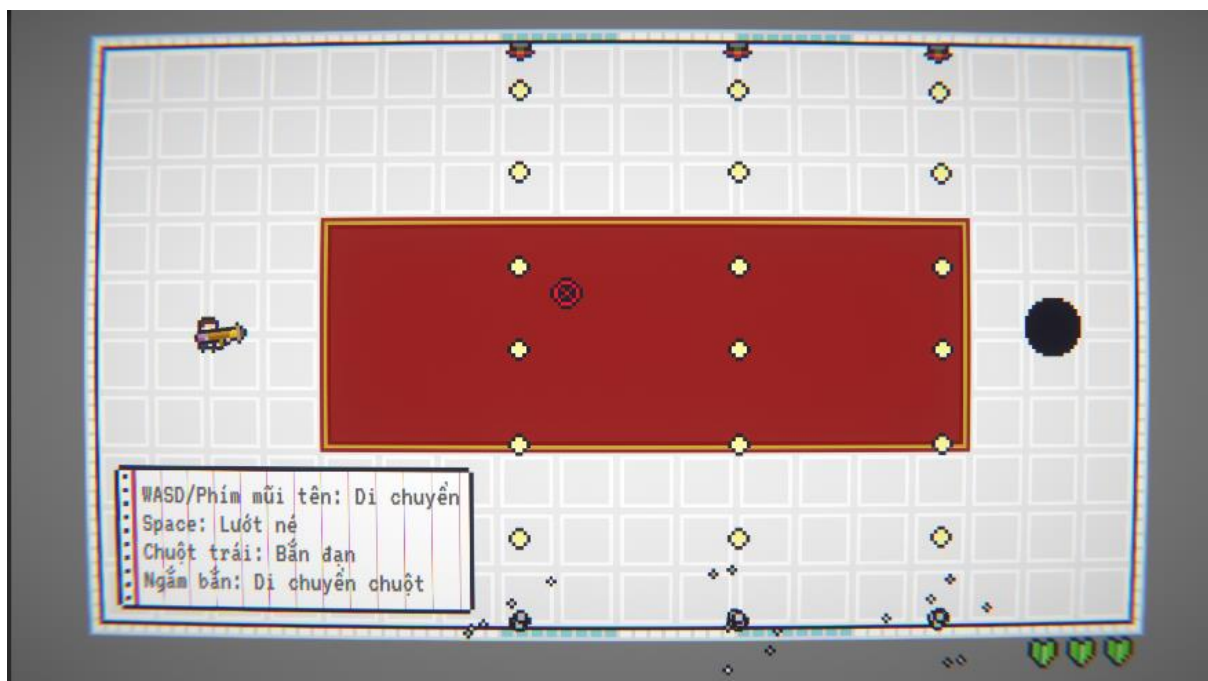
Khi bắt đầu trò chơi sẽ vào màn hình menu chính, sẽ có 3 nút tùy chọn cho người chơi là “Chơi mới”, “Cài đặt” và “Thoát”. Bấm nút “Chơi mới” để vào màn 1 của trò chơi, nút “Cài đặt” để chỉnh âm thanh và nút “Thoát” để thoát khỏi trò chơi.



Hình 4.1. Màn hình menu chính

4.1.2. Hướng dẫn trò chơi

Sau khi bấm vào nút chơi mới, người chơi sẽ được chuyển đến màn hướng dẫn chơi trò chơi. Nhân vật người chơi điều khiển ở góc trái màn hình. Mục tiêu của người chơi là đến được với cổng dịch chuyển nằm ở bên phải màn hình và tránh các ụ súng liên tục bắn đạn. Có một “tờ ghi chú” nằm ở góc trái dưới màn hình hướng dẫn người chơi các thao tác cơ bản như di chuyển, né tránh, nhắm bắn.



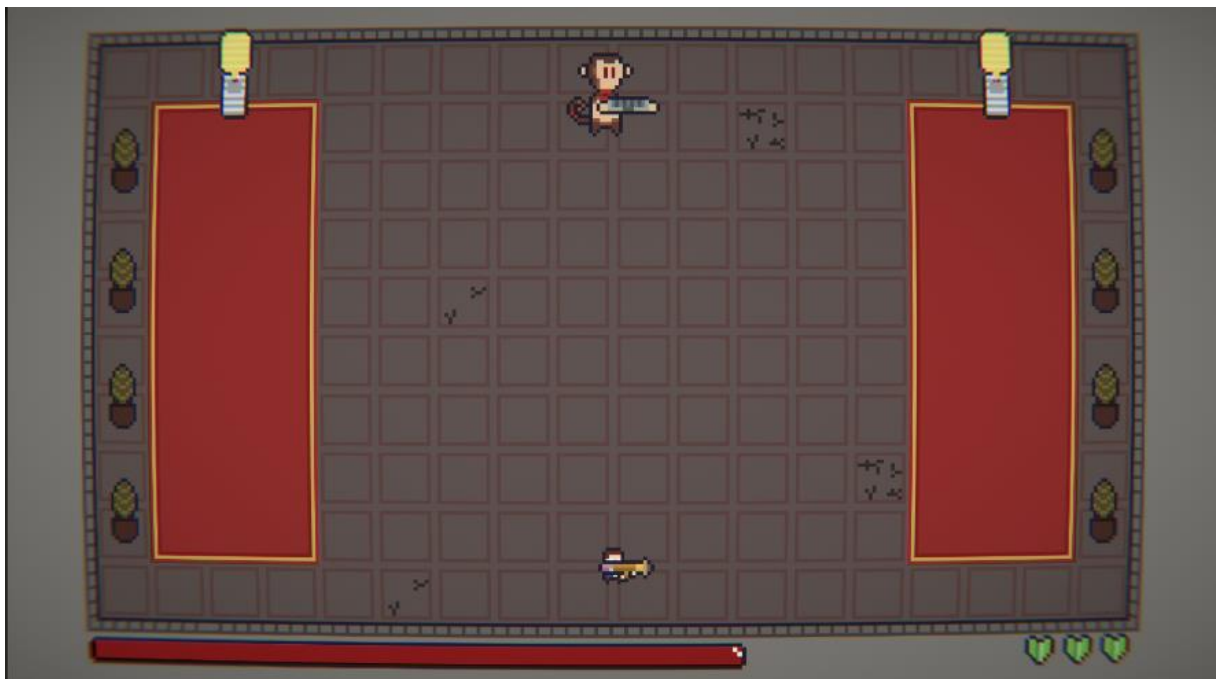
Hình 4.2. Màn hình hướng dẫn trò chơi

Góc phải dưới màn hình là số lượng máu của người chơi, bao gồm 3 máu. Mỗi lần dính sát thương người chơi sẽ mất 1 máu. Khi người chơi hết máu, trò chơi sẽ kết thúc. Màn hình GameOver hiện ra, cho người chơi lựa chọn chơi lại màn hoặc trở về menu chính.

4.1.3. Các màn chơi chính

Sau khi hoàn thành màn chơi hướng dẫn và đi vào cổng dịch chuyển, người chơi sẽ được đưa đến các hầm ngục, đánh bại kẻ địch và đi tiếp vào hầm ngục tiếp theo. Các hầm ngục này có cấu trúc tương tự nhau, là một căn phòng và một con trùm mà người chơi cần phải tiêu diệt để đi tiếp.

Con trùm sẽ có một thanh máu ở góc trái dưới màn hình. Khi người chơi gây sát thương lên trùm, thanh máu sẽ giảm theo số sát thương mà người chơi đã gây ra. Kẻ địch sẽ chết khi thanh máu giảm về 0, một cổng dịch chuyển sẽ xuất hiện đưa người chơi đến hầm ngục tiếp theo. Mỗi hầm ngục là một kẻ địch khác nhau, với lượng máu, cách di chuyển và tấn công khác nhau.



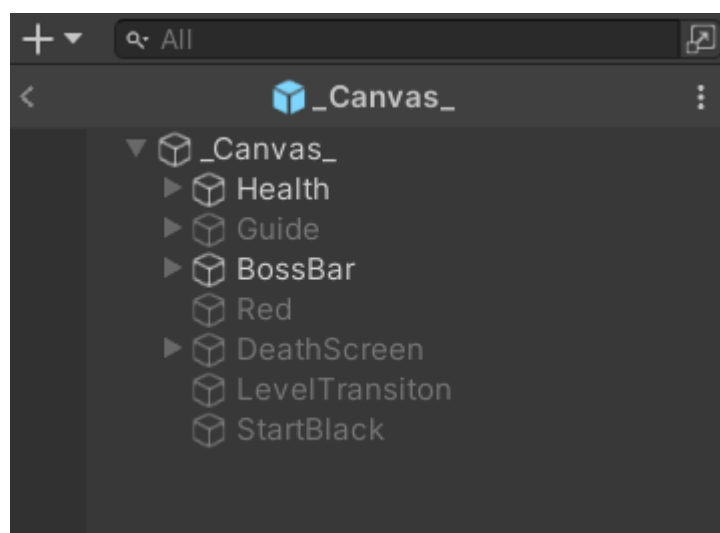
Hình 4.3 + 4.4. Một số màn chơi trong hầm ngục

4.2. Thiết kế trò chơi

4.2.1. Canvas

Canvas là một Prefabs – là một đối tượng được xây dựng sẵn và có thể tái sử dụng nhiều lần, thực hiện công việc hiển thị giao diện người dùng. Mỗi một màn chơi khác nhau ta có thể tái sử dụng và chỉnh sửa lại thành phần Canvas để hiển thị nội dung mong muốn. Các thành phần cơ bản của Canvas là:

- Hiển thị thanh máu người chơi
- Hiển thị thanh máu của kẻ địch
- Hiển thị màn hình GameOver
- Điều khiển màn chơi, hiệu ứng hình ảnh như chuyển màn, báo đỏ,...
- Hiển thị chữ, các nút bấm,...



Hình 4.5. Các thành phần trong Canvas

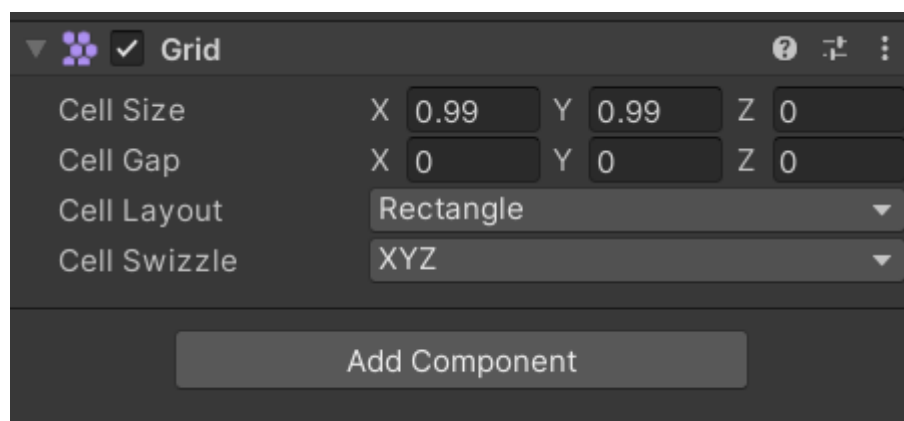
Các thành phần con bên trong Canvas có thể được tắt hoặc bật để hiển thị lên màn hình. Có thể điều khiển bật/tắt thông qua code để hiển thị các thành phần phù hợp như khi người chơi hết máu thì bật thành phần “DeathScreen” để hiển thị màn hình GameOver, khi người chơi còn 1 máu thì bật thành phần “Red” để hiển thị hiệu ứng đỏ màn hình,...



Hình 4.6. Màn hình GameOver

4.2.2. Background

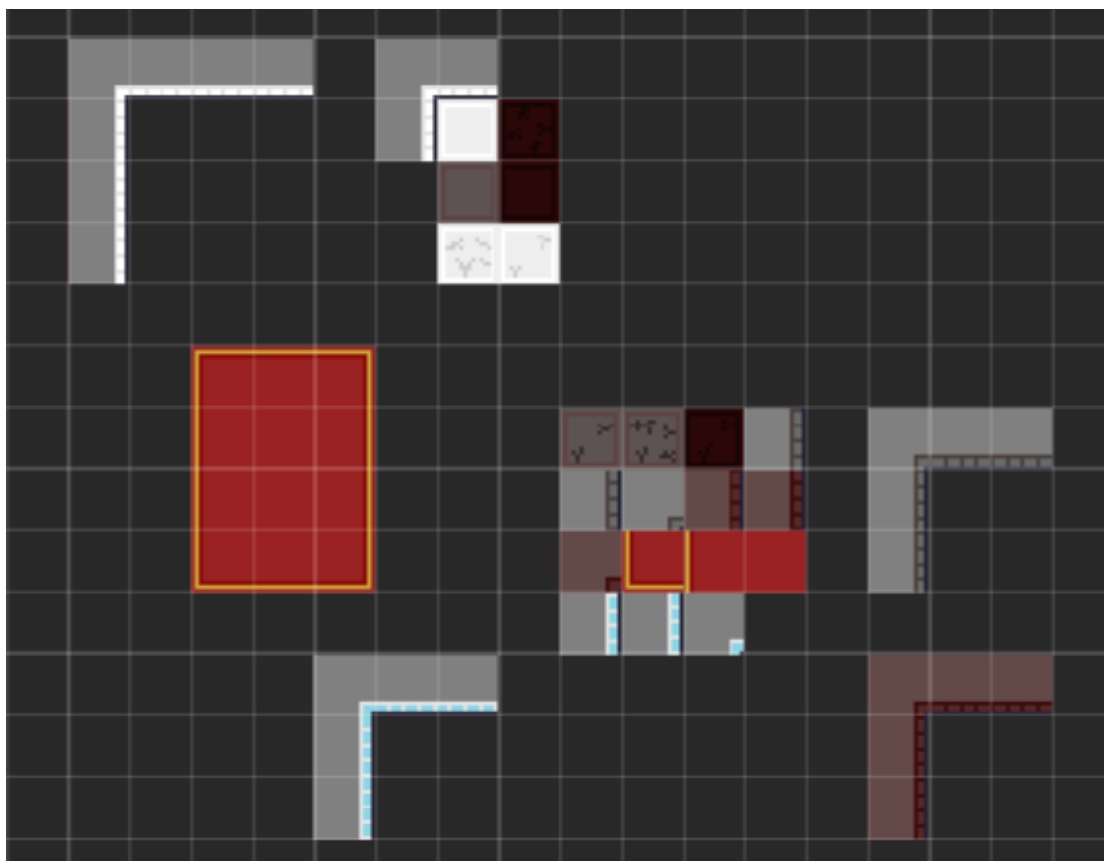
Các hàm ngục, background màn hình đều được thiết kế chia bảng và ô sử dụng Component có tên là “Grid”. Grid được sử dụng để chia màn hình thành các hàng và cột có thể chỉnh sửa kích thước. Từ đó ta có thể thiết kế màn chơi theo phong cách các khối vuông hình ảnh xếp vào nhau.



Hình 4.7. Component Grid

Trong thành phần Grid có 2 thành phần con là “Floor” và “Wall” để thiết kế sàn và tường của màn chơi. Sàn của màn chơi được tạo bằng công cụ “Tile Pallette”, giúp cắt hình ảnh có sẵn thành các hình vuông theo tỉ lệ cho trước và xếp các hình ảnh này vào các ô Grid. Tường của màn chơi là viền của sàn, sử dụng Component “Collider 2D” để tính toán va chạm và “Rigidbody 2D” ngăn người chơi không thể đi qua.

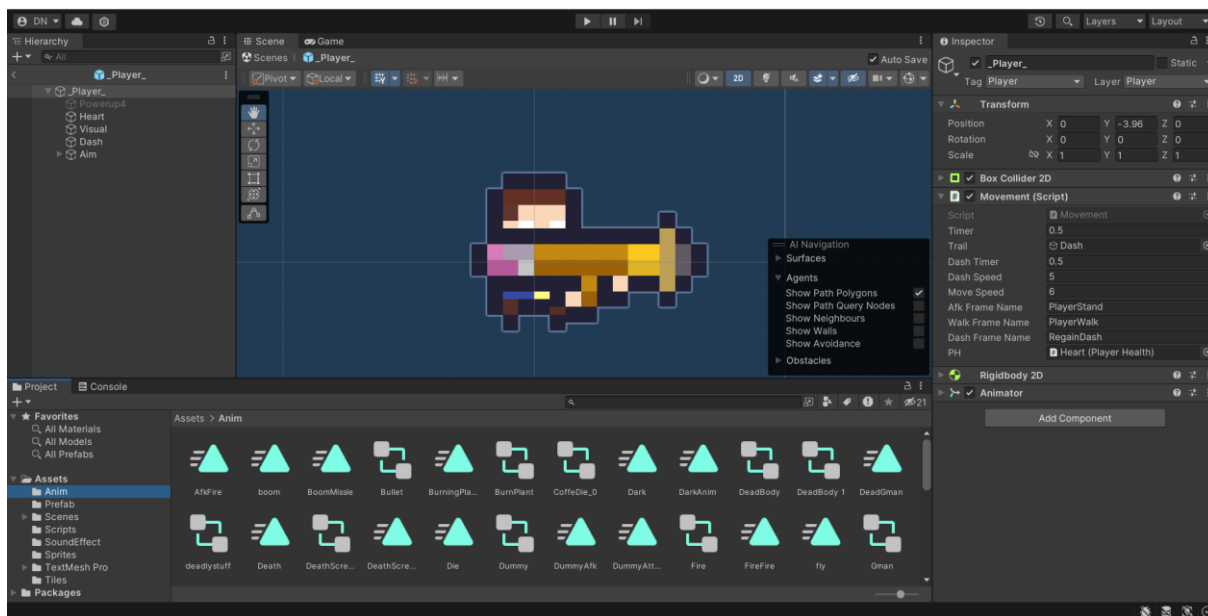
Hiệu ứng hình ảnh của trò chơi được thiết kế bị bóp méo để giả lập những trò chơi cổ điển. Component “Post-process Volume” được sử dụng để tạo ra hiệu ứng bóp méo màn hình theo hình cầu và hiệu ứng nhiễu sóng màn hình.



Hình 4.8. Hình ảnh đã được cắt để xếp vào các ô Grid

4.2.3. Player

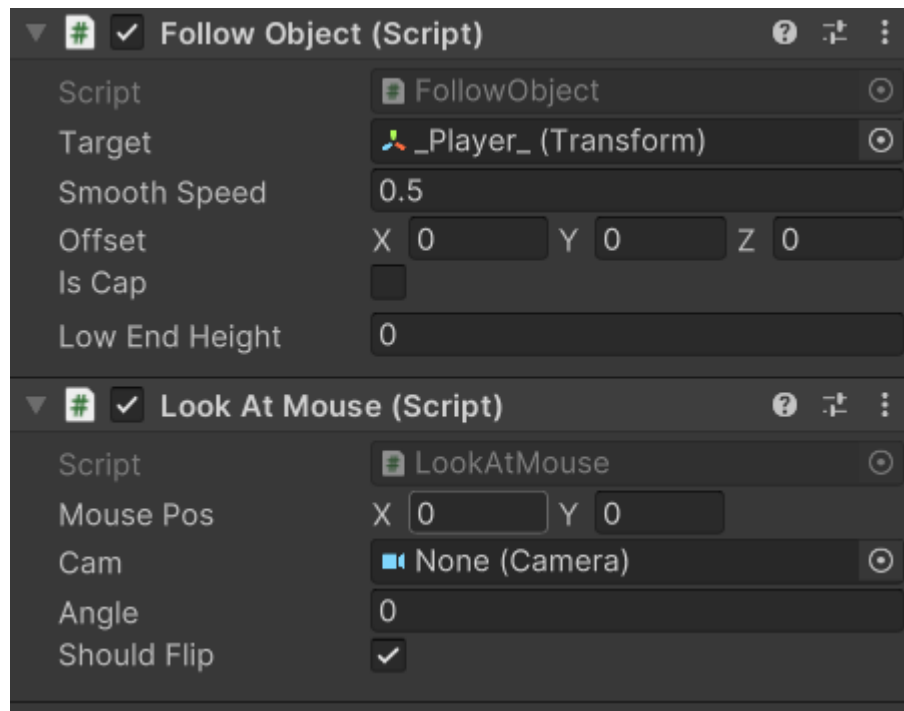
Player là nhân vật người chơi điều khiển, được xây dựng là một Prefabs để có thể tái sử dụng ở các màn chơi. Người chơi có thể điều khiển Player di chuyển theo các hướng trong mặt phẳng 2D, sử dụng chuột để ngắm và nhấn chuột trái để bắn đạn. Con trỏ chuột của máy tính được thay đổi thành hình tâm ngắm, giúp việc điều khiển trở nên trực quan và dễ dàng hơn.



Hình 4.9. Player

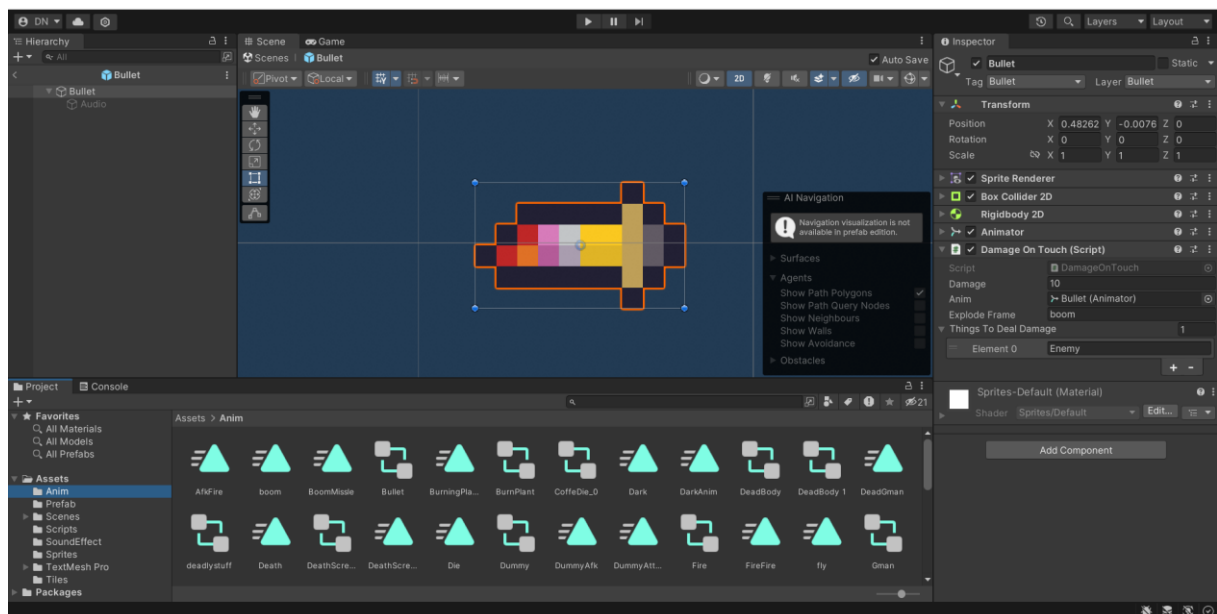
Ta sử dụng một biến bool là “Can be Damage” để làm cho nhân vật có thể bị dính sát thương hay không. Khi người chơi lướt né bằng nút Space, biến Can be Damage sẽ có giá trị False, nhân vật sẽ không nhận sát thương. Sau khi hoàn thành lướt né, biến này trở về giá trị True và một âm thanh “Tit” phát ra để báo hiệu người chơi có thể bị dính sát thương.

Thành phần “Aim” là thành phần điều khiển hoạt động nhắm bắn của Player. Nó sẽ liên tục lấy tọa độ của con trỏ chuột hiện tại để điều khiển hình ảnh của khẩu súng luôn chĩa về phía con trỏ chuột. Khi người dùng di chuyển chuột qua trái hoặc phải của nhân vật, hình ảnh Player sẽ đảo ngược qua phía tương ứng để diễn tả hoạt động đi qua trái hoặc phải của nhân vật.



Hình 4.10. Điều khiển súng luôn nhìn về phía con trỏ chuột

Viên đạn người chơi bắn ra cũng được xây dựng là một Prefabs, có các thành phần như âm thanh khi bắn và khi bị phá hủy, xử lý va chạm, các hiệu ứng hình ảnh và tính sát thương gây ra cho kẻ địch.



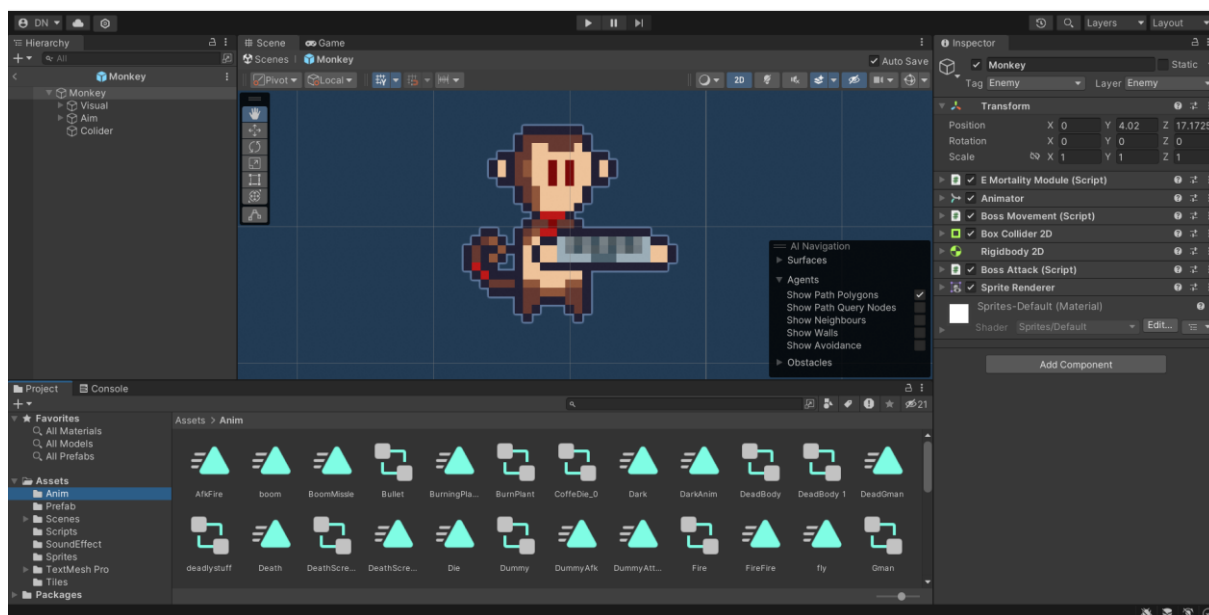
Hình 4.11. Viên đạn của người chơi

4.2.4. Kẻ địch

Mỗi hàm ngục sẽ có một kẻ địch khác nhau để người chơi đối đầu. Tuy chúng có hình dáng, cách di chuyển và tấn công khác nhau, nhưng chúng đều được xây dựng dựa trên các thành phần sau:

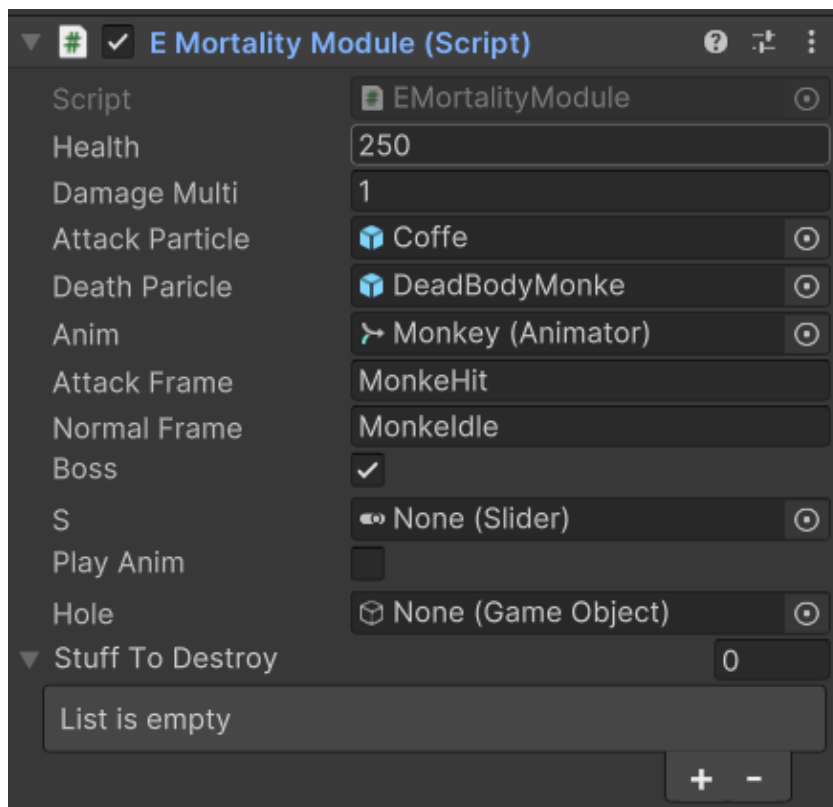
- Điều khiển va chạm, hoạt ảnh, hiệu ứng vật lý
- Điều khiển sự di chuyển
- Điều khiển lượng máu, còn sống hay đã bị tiêu diệt
- Điều khiển các đòn tấn công

Hiệu ứng hình ảnh là một phần không thể thiếu, chúng giúp tăng trải nghiệm người dùng và giúp trò chơi trở nên thú vị hơn. Các hiệu ứng hình ảnh được tạo nên từ những hình ảnh rời rạc, khi ghép chúng lại với một tốc độ nhất định sẽ tạo ra hoạt ảnh. Ta sử dụng Component Animator để đưa những hoạt ảnh đó vào những thời điểm phù hợp như đứng im hay di chuyển trái phải.



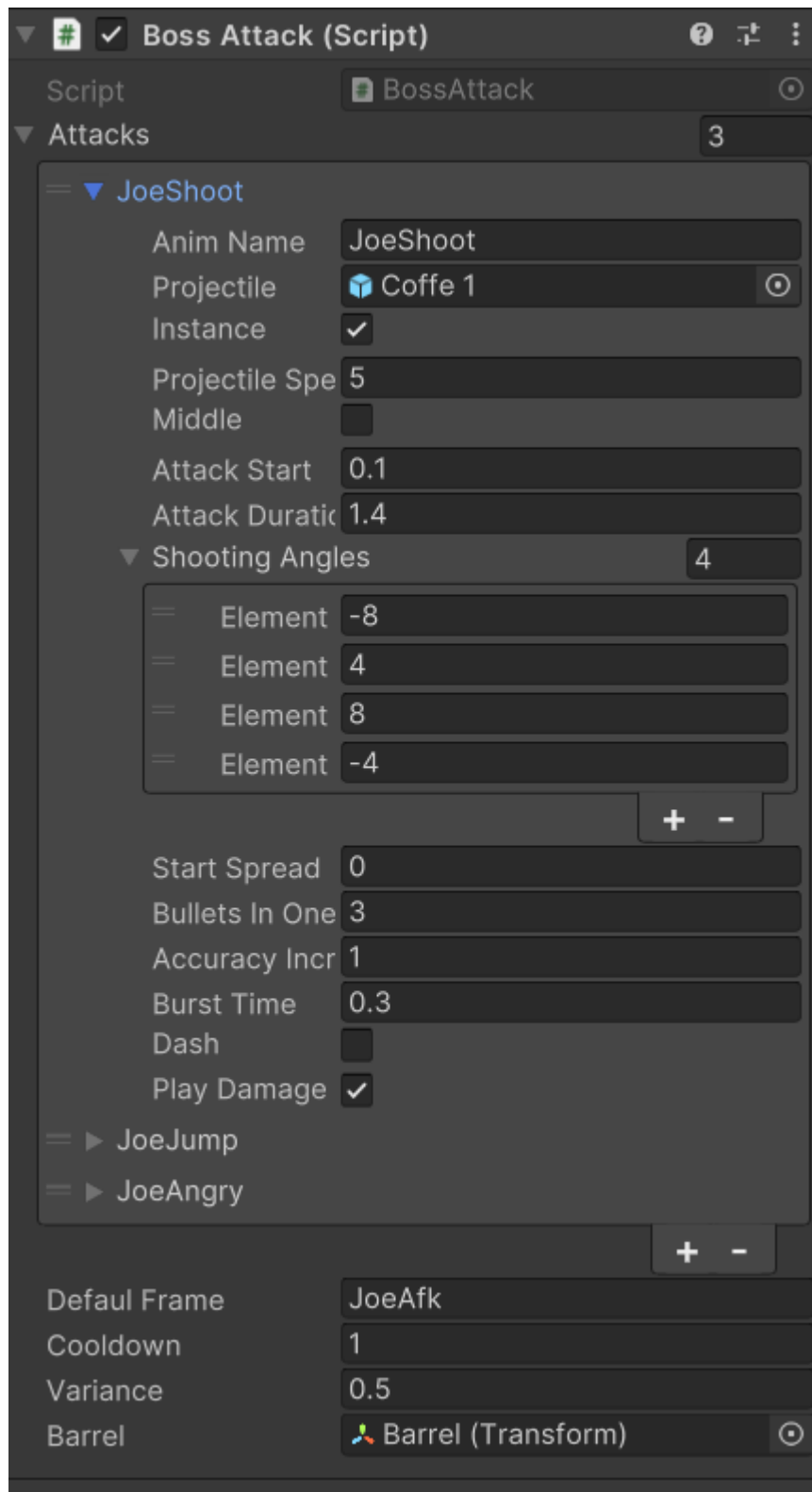
Hình 4.12. Kẻ địch trong trò chơi

Ta sử dụng Script “E Mortality Module” để điều khiển lượng máu, kiểm tra kẻ địch đã bị tiêu diệt hay chưa. Nếu kẻ địch đã bị tiêu diệt, hoạt ảnh tử trận sẽ được phát và một cổng dịch chuyển sẽ được xuất hiện gần đó. Người chơi đi vào cổng để đến màn tiếp theo.

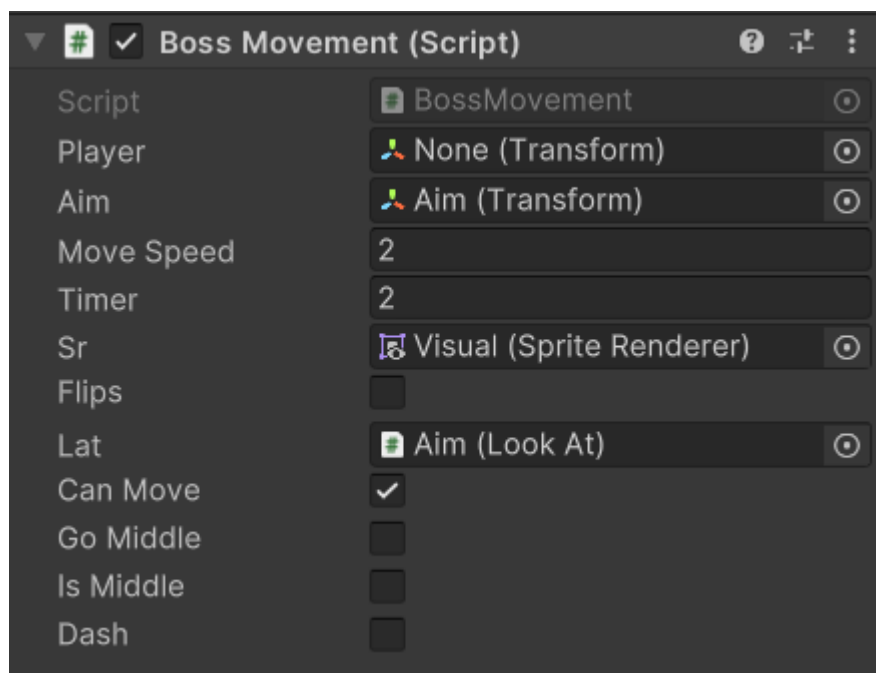


Hình 4.13. Script E Mortality Module

Để điều khiển các đòn tấn công của kẻ địch và khiến chúng tấn công một cách ngẫu nhiên, ta sử dụng script “Boss Attack”. Ta liệt kê ra một danh sách các đòn tấn công của kẻ địch, điều chỉnh các thông số như hình ảnh viên đạn bắn ra, hoạt ảnh tấn công, độ chính xác của viên đạn, số đạn bắn ra trong một lần tấn công,... Sau đó ta dùng thuật toán lấy ngẫu nhiên trong danh sách một đòn đánh và thi triển chúng. Sau mỗi lần thi triển đòn đánh, kẻ thù sẽ đợi một thời gian và tiếp tục lặp lại quá trình trên. Liệt kê danh sách đòn đánh khiến việc viết code và điều chỉnh thông số trở nên đơn giản, trực quan và dễ dàng sửa đổi.

*Hình 4.14. Script Boss Attack*

Script “Boss Movement” dùng để điều khiển sự di chuyển của kẻ địch. Ta khai báo biến `PlayerTransform` để liên tục cập nhật tọa độ của Player, từ đó kẻ địch sẽ bắn đạn về phía của nhân vật. Một số thông số khác như “`MoveSpeed`” để điều khiển tốc độ di chuyển của kẻ địch, “`GoMiddle`” để khiến kẻ địch đi ra giữa màn hình, dùng để thực hiện một đòn tấn công đặc biệt.

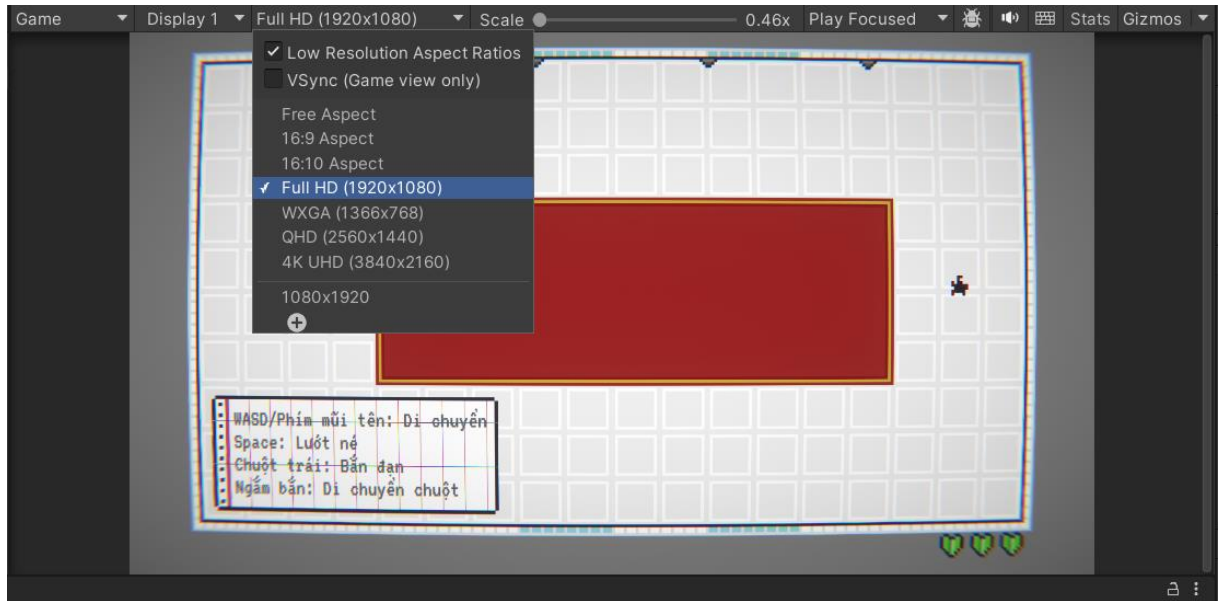


Hình 4.15. Script Boss Movement

4.3. Xây dựng trò chơi trên nền tảng máy tính

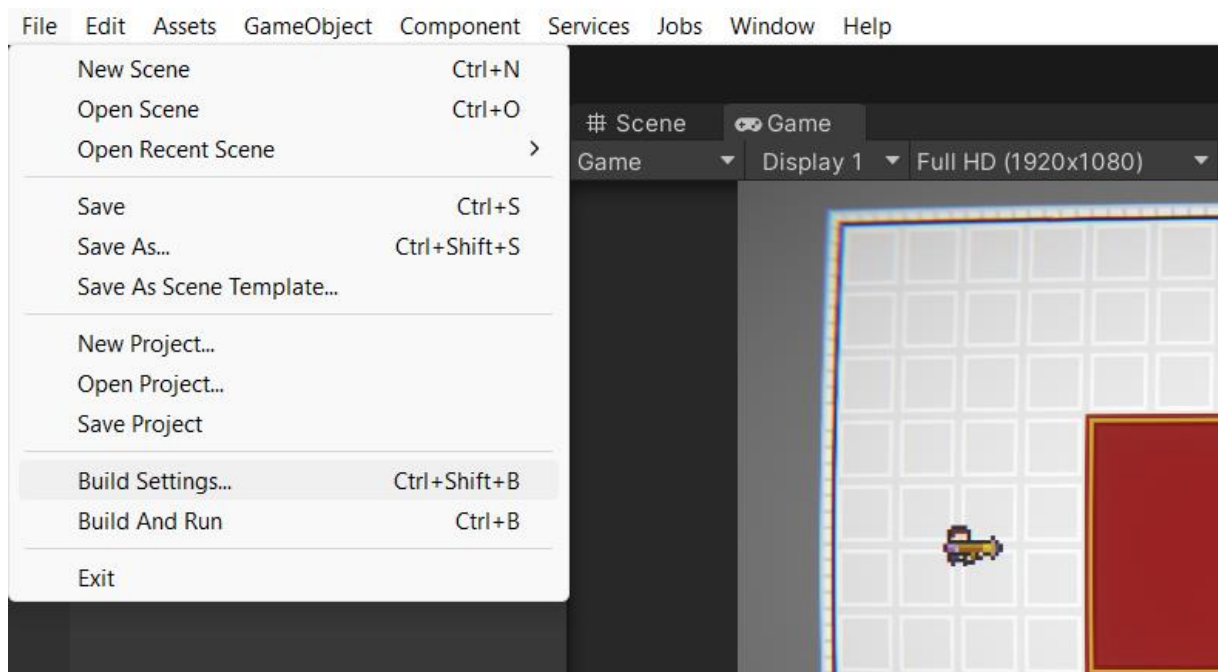
Sau khi đã hoàn thành công đoạn thiết kế và lập trình, bước cuối cùng để hoàn thiện trò chơi đến tay người dùng là xây dựng trò chơi trên nền tảng máy tính hoặc điện thoại, tùy vào hướng thiết kế của nhà phát triển. Unity là một Game Engine có thể hỗ trợ rất tốt việc xây dựng trò chơi trên đa nền tảng. Đồ án “Dungeon and Gun” được xây dựng trên nền tảng máy tính gồm các bước sau:

➤ **Bước 1:** Lựa chọn kích thước màn hình tiêu chuẩn (1920x1080)



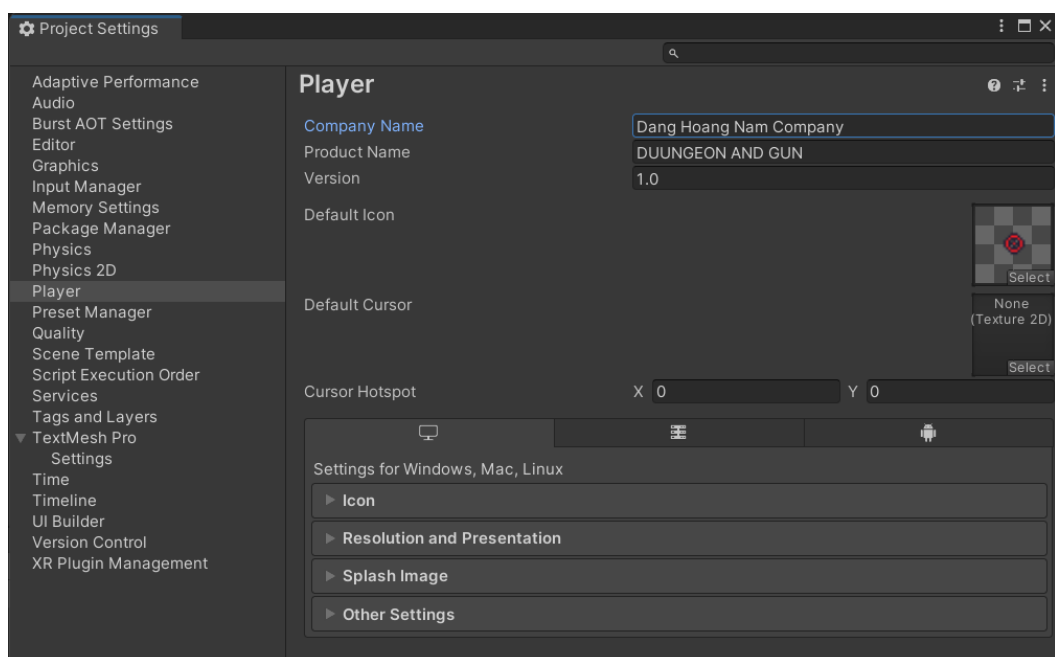
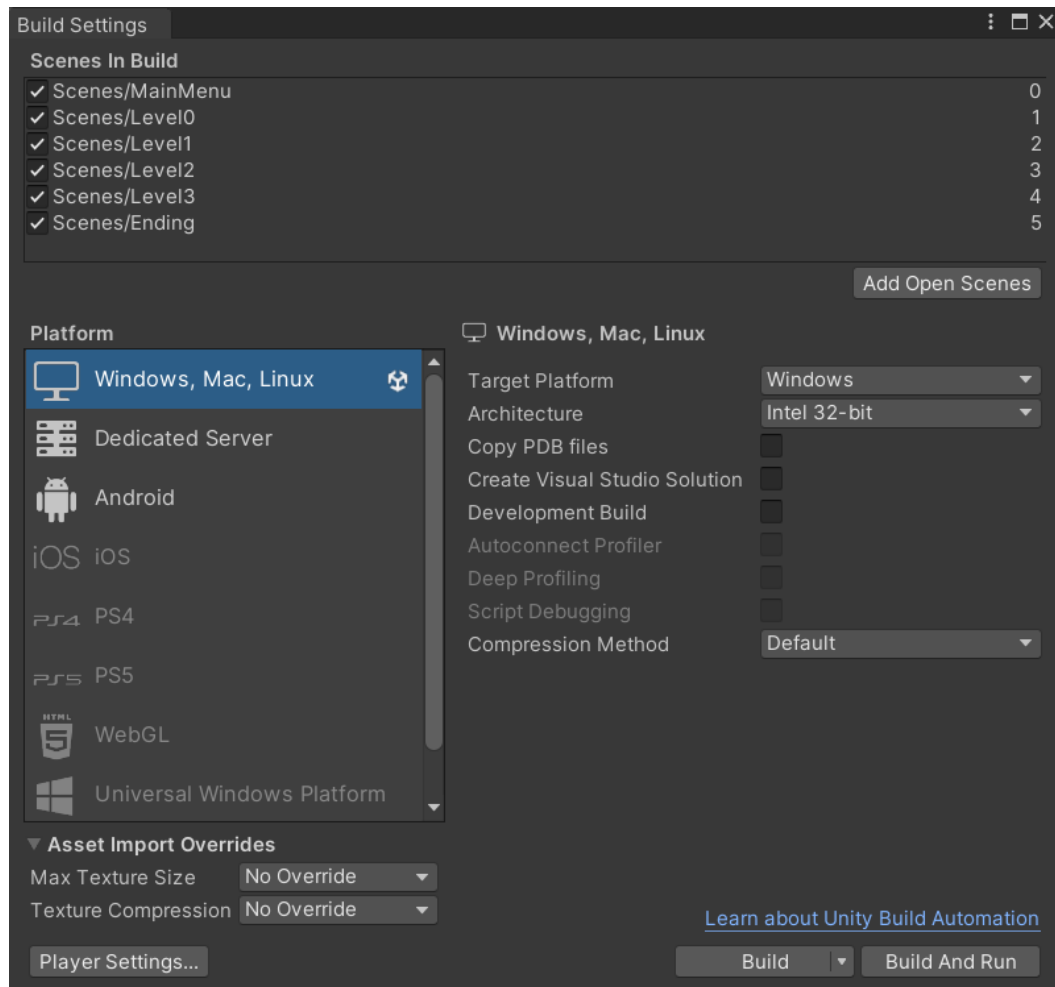
Hình 4.16. Lựa chọn kích thước màn hình

➤ **Bước 2:** Chọn File => Build Settings



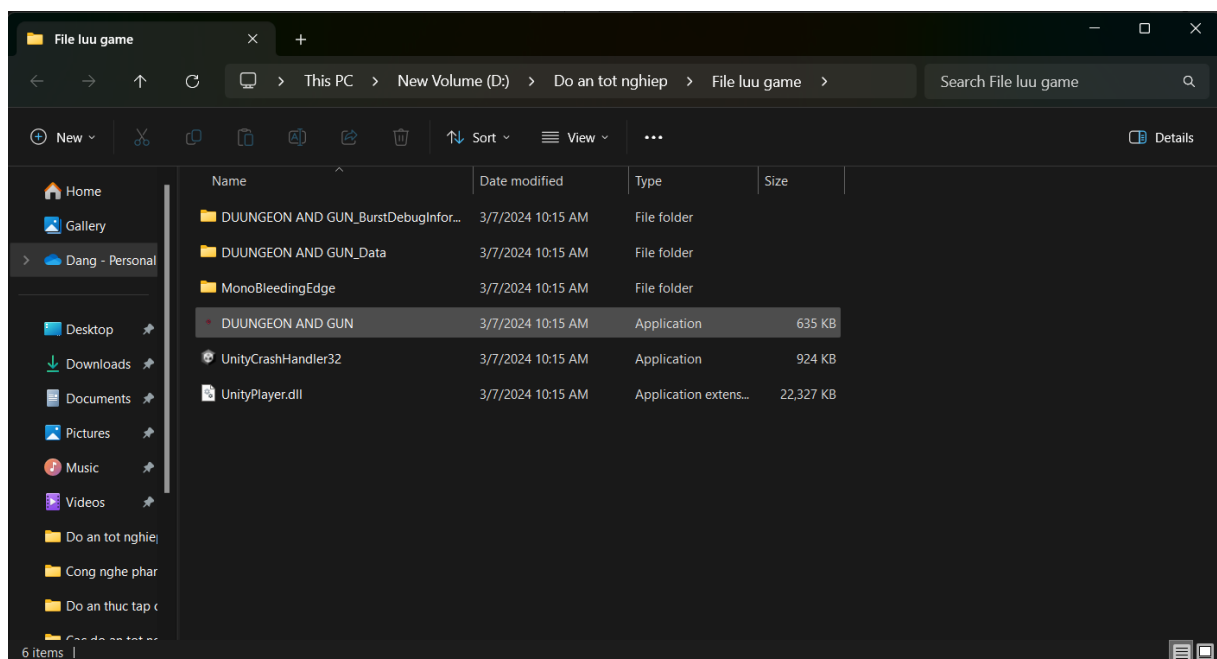
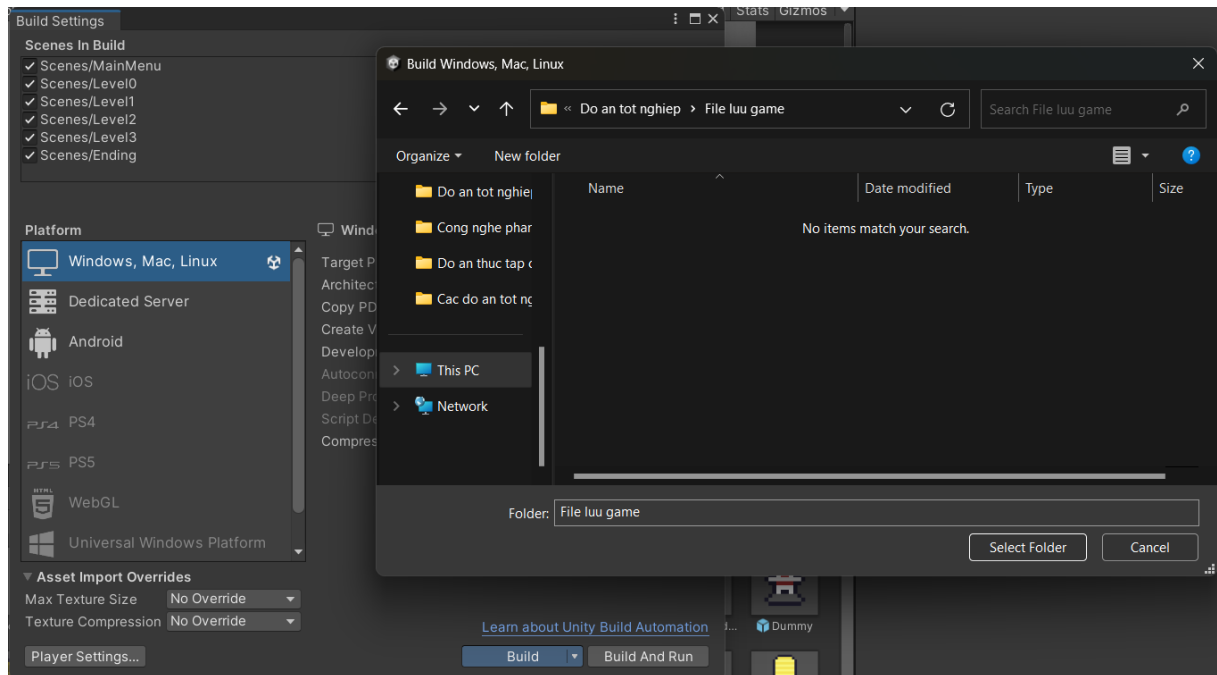
Hình 4.17. Chọn File => Build Settings

- **Bước 3:** Trong mục Build Settings, chọn mục Windows trong Platform, chọn Player Settings ở góc trái dưới màn hình để cài đặt tên và icon trò chơi



Hình 4.18 + 4.19. Build Settings và Player Settings

➤ Bước 4: Bấm Build, chọn nơi lưu file, hoàn thành xây dựng trò chơi



Hình 4.20 + 4.21. Hoàn thành xây dựng trò chơi

CHƯƠNG V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Nhận xét và kết luận

Cùng với sự nỗ lực của bản thân và sự hướng dẫn của cô **Trần Thị Thanh**, em đã hoàn thành đồ án tốt nghiệp “**Phát triển trò chơi Dungeon and Gun bằng phần mềm Unity**” ở tất cả các khâu phân tích, thiết kế, cài đặt và kiểm thử. Em nhận ra những ưu điểm và nhược điểm của đồ án như sau:

- **Ưu điểm:** Đồ án đã hoàn thiện các bước cơ bản như phân tích thiết kế, cài đặt kiểm thử, bước đầu tạo nên một trò chơi hoàn chỉnh. Trò chơi đáp ứng được những chức năng cơ bản, có dung lượng nhẹ, các lỗi của trò chơi hầu hết đã được xử lý.
- **Nhược điểm:** Việc phân tích vẫn còn chưa được trực quan. Chưa tối ưu các dòng lệnh, code còn rườm rà. Thời lượng chơi ngắn, chưa đa dạng về lối chơi, các loại vũ khí. Trò chơi chưa được tối ưu và xây dựng trên nền tảng điện thoại.

5.2. Hướng phát triển của đồ án

Trò chơi sẽ phát triển thêm các màn chơi mới, bổ sung thêm vũ khí mới, kẻ địch mới. Tối ưu thêm về thuật toán và giao diện, xây dựng trò chơi lên nền tảng thiết bị di động.

Trong quá trình làm đồ án, rất khó để tránh khỏi các sai sót không đáng có. Nhưng qua đây em đã rút ra rất nhiều kinh nghiệm cho bản thân. Em mong rằng đồ án này sẽ có ích và là một tài liệu tham khảo cho những bạn đam mê lập trình game. Em xin cảm ơn cô **Trần Thị Thanh** và các bạn đã giúp đỡ em trong quá trình nghiên cứu và hoàn thành đồ án tốt nghiệp này.

DANH MỤC HÌNH ẢNH

Hình 1.1. Trò chơi hành động bắn súng 2D

Hình 2.1. Unity Engine

Hình 2.2. Unity Assets Store

Hình 2.3. GameObject

Hình 2.4. Components

Hình 2.5. Prefabs

Hình 2.6. Camera

Hình 2.7. Scenes

Hình 2.8. Scripts

Hình 2.9. Microsoft Visual Studio

Hình 3.1. Biểu đồ Use-case

Hình 3.2. Biểu đồ hoạt động bắt đầu trò chơi

Hình 3.3. Biểu đồ điều khiển nhân vật

Hình 3.4. Biểu đồ tuần tự bắt đầu chơi

Hình 3.5. Biểu đồ tuần tự điều khiển nhân vật

Hình 3.6. Biểu đồ trạng thái

Hình 3.7. Biểu đồ thành phần

Hình 4.1. Màn hình menu chính

Hình 4.2. Màn hình hướng dẫn trò chơi

Hình 4.3 + 4.4. Một số màn chơi trong hàm ngục

Hình 4.5. Các thành phần trong Canvas

Hình 4.6. Màn hình GameOver

Hình 4.7. Component Grid

Hình 4.8. Hình ảnh được cắt để xếp vào các ô Grid

Hình 4.9. Player

Hình 4.10. Điều khiển súng luôn nhìn về phía con trỏ chuột

Hình 4.11. Viên đạn của người chơi

Hình 4.12. Kẻ địch trong trò chơi

Hình 4.13. Script E Mortality Module

Hình 4.14. Script Boss Attack

Hình 4.15. Script Boss Movement

Hình 4.16. Lựa chọn kích thước màn hình

Hình 4.17. Chọn File => Build Settings

Hình 4.18 + 4.19. Build Settings và Player Settings

Hình 4.20 + 4.21. Hoàn thành xây dựng trò chơi

TÀI LIỆU THAM KHẢO

- [1]. <https://learn.unity.com/>
- [2]. [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))
- [3]. <https://visualstudio.microsoft.com/>
- [4]. <https://itnavi.com.vn/blog/lap-trinh-unity>
- [5]. <https://codelearn.io/sharing/lap-trinh-game-voi-unity-nhu-the-nao-phan-1>
- [6]. <http://vietgamedev.net>
- [7]. <http://www.unity3dstudent.com/>