|  |  |
| --- | --- |
| BỘ GIÁO DỤC VÀ ĐÀO TẠO | BỘ NÔNG NGHIỆP VÀ PTNT |

TRƯỜNG ĐẠI HỌC THỦY LỢI



DƯƠNG GIÁP ĐỨC

**PHÁT TRIỂN TỰA GAME SINH TỒN 2D QUA CÔNG CỤ UNITY ENGINE**

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, NĂM 2023

|  |  |
| --- | --- |
| bỘ GIÁO DỤC VÀ ĐÀO TẠO | BỘ NÔNG NGHIỆP VÀ PTNT |

TRƯỜNG ĐẠI HỌC THỦY LỢI

DƯƠNG GIÁP ĐứC

**PHÁT TRIỂN TỰA GAME SINH TỒN 2D QUA CÔNG CỤ UNITY ENGINE**

|  |  |
| --- | --- |
| Ngành: | Công nghệ thông tin |
| Mã số: | 1951060614 |

|  |  |
| --- | --- |
| NGƯỜI HƯỚNG DẪN | 1. TS. Trương Xuân Nam |

HÀ NỘI, NĂM 2023

**GÁY BÌA ĐỒ ÁN TỐT NGHIỆP, KHÓA LUẬN TỐT NGHIỆP**

**DƯƠNG GIÁP ĐỨC ĐỒ ÁN/KL TỐT NGHIỆP HÀ NỘI, NĂM 2023**

**LỜI CAM ĐOAN**

Tác giả xin cam đoan rằng Đồ án tốt nghiệp là kết quả nghiên cứu của bản thân em và được sự hướng dẫn của TS. Trương Xuân Nam. Các kết quả trong đồ án tốt nghiệp đều là trung thực không qua bất kỳ nguồn nào và sao chép dưới bất kỳ hình thức nào. Việc tham khảo (nếu có) đã được thực hiện trích dẫn và ghi nguồn tài liệu tham khảo đúng quy định.

TÁC GIẢ ĐATN

Dương Giáp Đức

LỜI CÁM ƠN

Trong quá trình hoàn thiện đề tài “Phát triển game sinh tồn 2D qua công cụ Unity Engine”, em đã nhận được rất nhiều sự giúp đỡ của thầy cô và bạn bè.

Đầu tiên em xin chân thành cảm ơn thầy cô tại trường Đại học Thủy Lợi nói chung và các thầy cô trong khoa Công nghệ thông tin nói riêng đã chỉ dạy nên em có thể đúc kết lại những kiến thức học được để hoàn thiện đồ án này.

Tiếp đó em xin gửi lời cảm ơn sâu sắc tới TS. Trương Xuân Nam, cảm ơn thầy đã hướng dẫn tận tình em và góp ý cho em trong quá trình hoàn thiện đồ án này.

Với khả năng còn hạn chế trong đề tài được chọn không thể thiếu đi sự sai sót nhưng em tự tin rằng nhờ sự chỉ bảo của thầy cô nên em đã có thể hoàn thiện đồ án này tốt nhất có thể.

Em xin chân thành cảm ơn!

# MỤC LỤC

[LỜI CAM ĐOAN 1](#_Toc154878809)

[MỤC LỤC 3](#_Toc154878810)

[DANH MỤC HÌNH ẢNH 5](#_Toc154878811)

[TÓM TẮT ĐỀ TÀI 7](#_Toc154878812)

[CHƯƠNG 1 : TỔNG QUÁT VỀ NGÀNH CÔNG NGHIỆP GAME 12](#_Toc154878813)

[1.1 Lịch sử 12](#_Toc154878814)

[1.2 Một số công cụ làm game 13](#_Toc154878815)

[1.2.1 Godot 13](#_Toc154878816)

[1.2.2 GameMaker Studio 14](#_Toc154878817)

[1.2.3 RPG Maker. 15](#_Toc154878818)

[1.2.4 Unreal Engine: 17](#_Toc154878819)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT 19](#_Toc154878820)

[2.1 Unity Engine 19](#_Toc154878821)

[2.1.1 Tổng quát về Unity Engine 19](#_Toc154878825)

[2.1.2 Lý do sử dụng Unity Engine 19](#_Toc154878826)

[2.1.3 Màn hình làm việc chính của Unity và các thuộc tính cơ bản của Unity 20](#_Toc154878827)

[2.1.4 Các API sử dụng chính trong Unity 26](#_Toc154878828)

[2.2 Các công cụ hỗ trợ khác 34](#_Toc154878829)

[2.2.1 Visual Studio 34](#_Toc154878830)

[2.2.2 GIMP 34](#_Toc154878831)

[2.2.3 Github 35](#_Toc154878832)

[CHƯƠNG 3 : LÊN Ý TƯỞNG VÀ THIẾT KẾ GAME 36](#_Toc154878833)

[3.1. Lên ý tưởng 36](#_Toc154878834)

[3.2. Thiết kế game 36](#_Toc154878835)

[3.2.1. Bản đồ 36](#_Toc154878836)

[3.2.2. Nhân vật chính 37](#_Toc154878837)

[3.2.3. Quái vật 38](#_Toc154878838)

[3.2.4. Hệ thống vũ khí 38](#_Toc154878839)

[3.2.5. Hệ thống cấp độ và nâng cấp 38](#_Toc154878840)

[3.2.6. Điều khiển 39](#_Toc154878841)

[3.2.7. Các hệ thống khác 39](#_Toc154878842)

[CHƯƠNG 4 : PHÂN TÍCH HỆ THỐNG 40](#_Toc154878843)

[4.1 4.1. Phân tích các actor chính 40](#_Toc154878844)

[4.2 4.2. Xây dựng biểu đồ usercase 40](#_Toc154878845)

[4.2.1 4.2.1. Mô hình usercase tổng quát 41](#_Toc154878846)

[4.3 4.3. Danh sách các usercase 41](#_Toc154878847)

[TÀI LIỆU THAM KHẢO 43](#_Toc154878848)

[CÁC ASSET SỬ DỤNG TRONG ĐỀ TÀI 44](#_Toc154878849)

# DANH MỤC HÌNH ẢNH

[Hình 1 Logo của Godot 14](#_Toc155131219)

[Hình 2 Màn hình làm việc của Godot 14](#_Toc155131220)

[Hình 3 Logo của Game Maker 15](#_Toc155131221)

[Hình 4 Màn hình làm việc của GameMaker 15](#_Toc155131222)

[Hình 5 Logo của RPG Maker 16](#_Toc155131223)

[Hình 6 Màn hình làm việc của RPG Maker 16](#_Toc155131224)

[Hình 7 Logo của Unreal Engine 17](#_Toc155131225)

[Hình 8 Màn hình làm việc của Unreal Engine 17](#_Toc155131226)

[Hình 9 Logo của Unity 18](#_Toc155131227)

[Hình 10 Màn hình làm việc chính của Unity 19](#_Toc155131228)

[Hình 11 Một số GameObject 20](#_Toc155131229)

[Hình 12 Di chuyển lớp Player 21](#_Toc155131230)

[Hình 13 Hình ảnh Inspector của lớp Player 21](#_Toc155131231)

[Hình 14 Cấu trúc của folder trong Project Window 22](#_Toc155131232)

[Hình 15 Scene View bao gồm GameObject Player và các thuộc tính 22](#_Toc155131233)

[Hình 16 Game View màn hình trò chơi cuối cùng mà người chơi sẽ trải nghiệm 23](#_Toc155131234)

[Hình 17 Ví dụ về một GameObject 23](#_Toc155131235)

[Hình 18 Transform của lớp Player trên màn hình 24](#_Toc155131236)

[Hình 19 Ví dụ về sử dụng Vector để tạo ra di chuyển 24](#_Toc155131237)

[Hình 20 Ví dụ về một collider 25](#_Toc155131238)

[Hình 21 Lớp Rigidbody2D khi thêm vào GameObject Player 25](#_Toc155131239)

[Hình 22 Hình ảnh của script GameManager 26](#_Toc155131240)

[Hình 23 Hình ảnh UI của game 27](#_Toc155131241)

[Hình 24 Ví dụ về một script sử dụng scripable object 28](#_Toc155131242)

[Hình 25 Scriptable Object tạo từ editor 28](#_Toc155131243)

[Hình 26 Ví dụ về một script MonoBehavior khi được khởi tạo 29](#_Toc155131244)

[Hình 27 Hình ảnh cấu trúc của bản đồ 31](#_Toc155131245)

[Hình 28 Sprite nhân vật chính 32](#_Toc155131246)

[Hình 29 Hình ảnh quái vật Dơi 33](#_Toc155131247)

[Hình 30 Màn hình lên cấp và hệ thống nâng cấp 33](#_Toc155131248)

[Hình 31 Biểu đồ usercase tổng quát 41](#_Toc155131249)

[Hình 32 Biểu đồ usercase di chuyển 42](#_Toc155131250)

[Hình 33 Biểu đồ hoạt động di chuyển 43](#_Toc155131251)

[Hình 34 Biểu đồ usercase AI của quái vật 44](#_Toc155131252)

[Hình 35 Biểu đồ hoạt động AI của quái vật 45](#_Toc155131253)

[Hình 36 Biểu đồ usercase hệ thống vũ khí 46](#_Toc155131254)

[Hình 37 Biểu đồ hoạt động của hệ thống vũ khí 46](#_Toc155131255)

[Hình 38 Biểu đồ usercase lên cấp 47](#_Toc155131256)

[Hình 39 Biểu đồ hoạt động lên cấp 48](#_Toc155131257)

[Hình 40 Biểu đồ usercase nâng cấp 49](#_Toc155131258)

[Hình 41 Biểu đồ hoạt động nâng cấp 50](#_Toc155131259)

[Hình 42 Biểu đồ usercase xử lý va chạm người chơi 51](#_Toc155131260)

[Hình 43 Biểu đồ hoạt động xử lý va chạm của người chơi 52](#_Toc155131261)

[Hình 44 Biểu đồ usercase xử lý va chạm của quái vật 53](#_Toc155131262)

[Hình 45 Biểu đồ hoạt động xử lý va chạm của quái vật 54](#_Toc155131263)

# DANH MỤC BẢNG BIỂU

[Bảng 1 Sơ đồ usercase di chuyển của người chơi 37](#_Toc155131299)

[Bảng 2 Usercase di chuyển của quái vật 38](#_Toc155131300)

[Bảng 3 Usercase hệ thống vũ khí và nội tại 38](#_Toc155131301)

[Bảng 4 Usercase lên cấp 39](#_Toc155131302)

[Bảng 5 Usercase nâng cấp 40](#_Toc155131303)

[Bảng 6 Usercase xử lý va chạm của người chơi 40](#_Toc155131304)

[Bảng 7 Usercase xử lý va chạm của quái vật 41](#_Toc155131305)

|  |  |
| --- | --- |
| logo | TRƯỜNG ĐẠI HỌC THUỶ LỢI  **KHOA CÔNG NGHỆ THÔNG TIN**  BẢN TÓM TẮT ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP |

TÊN ĐỀ TÀI: **Phát triển tựa game sinh tồn 2D qua công cụ Unity Engine**

Sinh viên thực hiện: Dương Giáp Đức

*Lớp:* 61TH1

*Mã sinh viên:* 1951060614

*Giáo viên hướng dẫn:* ThS. Trương Xuân Nam.

# TÓM TẮT ĐỀ TÀI

Vào thời kì phát triển của công nghệ, đi kèm với đó là những loại hình giải trí như trò chơi điện tử cũng qua đó càng ngày càng phát triển hơn. Nhắm vào thị trường đang trên đà phát triển đó để mang lại một tựa game có tính thử thách, có khả năng chơi lại cao và hơn hết là mang tính chất giải trí cho người chơi. Qua đó tựa game sinh tồn 2D được ra mắt để đáp ứng những tiêu chí trên đánh vào thị yếu những game thủ thích game sinh tồn.

Game sinh tồn 2D này có sự tương đồng và lấy ý tưởng từ tựa game cùng thể loại cực kỳ nổi tiếng qua một vài năm trở lại đây là Vampire Survivor, thay vì phát triển các lớp nhân vật có khởi đầu vũ khí khác nhau, tựa game sẽ chỉ có một nhân vật và khi bắt đầu được người chơi sẽ lựa chọn một trong các vũ khí khởi đầu, sinh tồn qua một khoảng thời gian nhất định, hệ thống giết kẻ địch để thăng cấp và nâng cấp các vũ khí, kĩ năng.

Về gameplay chính của tựa game, game sinh tồn 2D sẽ cho phép người chơi di chuyển theo 4 hướng, camera sẽ chạy theo nhân vật. Với một số vũ khí bắn thì sẽ được thiết kế tự động bắn theo hướng người chơi đang di chuyển mà không phải thông qua chuột, giết địch lên cấp và nâng cấp các vũ khí để tăng hiệu quả trong quá trình giết quái. Quái vật sẽ được sinh ra với số lượng nhất định, có thể giết hết chúng rồi hoàn thiện màn chơi hoặc sinh tồn trong một khoảng thời gian nhất định.

Hướng mà trò chơi hướng đến là các nâng cấp sẽ giúp người chơi mang lại sự thỏa mãn khi tiêu diệt các quái vật, mang đến cho người chơi một khoảng thời gian giải trí nhẹ nhàng và ngắn hạn.

Với những đặc điểm trên và muốn thử sức với một đề tài trong ngành công nghiệp mà bản thân cũng tiêu thụ rất nhiều, em đã chọn đề tài trên để nắm rõ hơn về ngành công nghiệp làm game nói chung và quy trình phát triển một tựa game.

**Công nghệ sử dụng**

* Unity Engine
* Visual Studio
* Github
* GIMP

**Các mục tiêu chính**

Mục tiêu:

* Nhân vật di chuyển theo 4 hướng, camera sẽ chạy nhân vật.
* Các vũ khí bắn sẽ được bắn theo hướng di chuyển của nhân vật.
* Xây dựng cơ chế tự sinh bản đồ khi người chơi di chuyển.
* Tự động ngẫu nhiên các đồ vật trên khung cảnh của một tấm map sinh ra để tránh sự trùng lặp.
* Tạo ra các đồ vật ở môi trường có thể tương tác và rơi ra vật phẩm.
* Tạo ra AI đơn giản cho các quái vật.
* Tạo ra hệ thống chỉ số cho người chơi và quái vật.
* Tạo ra hệ thống cấp độ.
* Tạo ra hệ thống vũ khí và nội tại.
* Tạo ra hệ thống nâng cấp vũ khí, nội tại khi lên cấp.
* Tạo ra các luồng dừng, thoát game, màn hình hiển thị các chỉ số và nâng cấp trong và sau khi hoàn thiện màn chơi (hoặc chết).

Kỹ năng:

* Sinh viên tìm hiểu về kiến thức C#, các chức năng Unity Engine.
* Sinh viên thành công xây dựng một demo với các tính năng trên.

**Kết quả dự kiến**

Kiến thức:

* Sinh viên tìm hiểu về các kiến thức của C#, Unity Engine.
* Sinh viên nắm được cơ bản quy trình phát triển một tựa game.

Kỹ năng:

* Báo cáo tổng hợp về công cụ Unity Engine.
* Sinh viên lập trình thành công demo dựa trên các bài toán đề ra bằng công Unity Engine với ngôn ngữ chủ đạo là C#.

**Bố cục của báo cáo**

* Chương 1: Tổng quát về ngành công nghiệp game
* Chương 2: Cơ sở lý thuyết
* Chương 3: Ý tưởng và hướng thiết kế game
* Chương 4: Phân tích hệ thống
* Chương 5: Xây dựng và demo các chức năng
* Chương 6: Kết luận.

# : TỔNG QUÁT VỀ NGÀNH CÔNG NGHIỆP GAME

## Lịch sử

Ngành công nghiệp game là một trong những ngành thuộc ngành giải trí. Phát triển từ những thứ thô sơ nhất như những món đồ chơi hay những board game (ví dụ như cờ tỷ phú) sau đó phát triển dần hơn đến với định nghĩa “trò chơi điện tử” khi mang đến những trò chơi với lối chơi tương tự như X, O lên trên các hệ thống máy vi tính đầu tiên, sau đó là đến thời kỳ của những máy game thùng với các tựa game như Space War (1962), Periscope (1966), . . . Nhờ những phát kiến và sự phát triển của thời đại công nghệ, nhu cầu sử dụng máy tính cá nhân cũng như sự phát triển của phần cứng máy tính ngành công nghiệp trò chơi điện tử ngày càng phát triển hơn nữa khi mang lại những trò chơi càng ngày càng có độ chân thực cao hơn và mang lại doanh thu lớn khi tính đến năm 2018 đã thu lại 134.9 tỷ đô la mỹ doanh thu toàn cầu do Entertainment Software Association thống kê. Nhờ vào đó mà cũng sản sinh ra các ngành nghề tương ứng như lập trình game, các nhà phát hành, đại lý bán game vật lý.

Ở Việt Nam cũng được trải qua những thời kỳ trên khi các tựa máy chơi game tay cầm PS1, PS Vita được mang về Việt Nam, sau đó là thời kỳ phát triển của các trò chơi trực tuyến như Võ Lâm Truyền Kỳ, Boom Online, . . . Bởi những nhà phát hành như VNG (tiền thân là Vinagame), VTC Game, . . . Sau một khoảng thời gian dài các nhà phát hành Việt Nam thay vì những tựa game phát hành trên máy tính, chủ yếu sẽ mang những tựa game trên điện thoại để phù hợp với thị yếu người dùng do sự phát triển của phần cứng điện thoại qua đó cũng thúc đẩy hơn ngành công nghiệp game điện thoại. Dù ngành công nghiệp game có phát triển như vậy nhưng đa phần những sản phẩm trên đều là các sản phẩm nước ngoài được phát hành tại Việt Nam chứ không phải tựa game chính thống do Việt Nam tạo ra. Ta có thể thấy rằng việc lập trình viên game ở Việt Nam vẫn là một ngành nghề rất mơ hồ nhưng không phải vì thế mà chúng ta không có các tựa game do người Việt Nam tạo ra ví dụ như 7554 (2011) dù không thành công nhưng cũng là một sản phẩm game bắn súng góc nhìn thứ nhất trên cửa hàng quốc tế hay một ví dụ cực kỳ nổi tiếng về game trên điện thoại là Flappy Bird (2013) với 90 triệu lượt tải về trên các nền tảng Google Play và AppStore là những lát gạch đầu tiên về sản phẩm của người Việt trên thị trường quốc tế. Hiện nay khi công nghệ đã phát triển hơn so với những năm trở về trước, những nhà phát hành game như Steam và Epic Store được ra đời giúp những nhà phát triển game dễ dàng mang lại tựa game của mình đến với người dùng hơn, hay các sản phẩm cần một khoảng đầu tư lớn thường hay gọi vốn trên KickStarter. Phải nói rằng hiện tại năm 2023 và tiếp đến chúng ta sẽ càng ngày càng thấy ngành công nghiệp game phát triển.

## Một số công cụ làm game

Nhờ vào sự phát triển của công nghệ phần mềm hiện tại, trên thì trường ngày nay có rất nhiều công cụ giúp đỡ các nhà phát triển dễ dàng hơn trong việc sáng tạo các trò chơi theo ý muốn của bản thân mình. Qua đây là một số những phần mềm khá là nổi tiếng trong giới làm game.

### Godot

Là một phần mềm mã nguồn mở miễn phí viết bằng ngôn ngữ C+ ra mắt vào năm 2014, hỗ trợ làm game trên mọi nền tảng PC, Android, Web. Điều thú vị của Godot là việc viết code cho game sử dụng một ngôn ngữ tích hợp riêng của Godot được gọi là GDScript, có sự tương tự với ngôn ngữ lập trình python và là một ngôn ngữ bậc cao.

A video game logo with a blue face

Description automatically generated with medium confidence

Hình 1 Logo của Godot

A screenshot of a computer

Description automatically generated

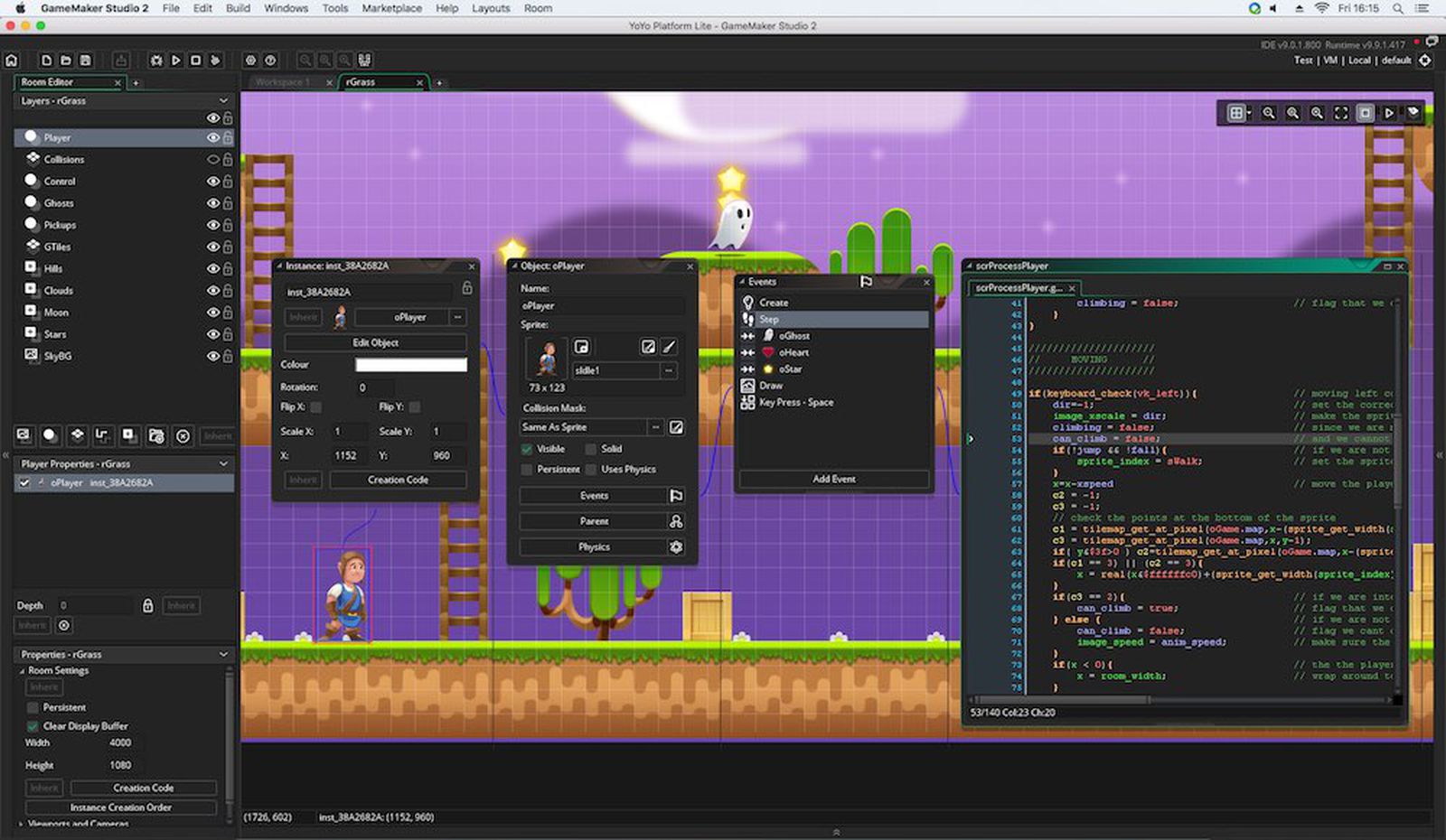
Hình 2 Màn hình làm việc của Godot

### GameMaker Studio

GameMaker tiền thân là Animo, là một phần mềm đa nền tảng viết bằng ngôn ngữ C++ được tạo ra từ năm 1999. GameMaker là một công cụ chủ yếu để thiết kế ra các tựa game 2D, dù nó có thể thiết kế các tựa game 3D nhưng còn rất nhiều hạn chế. Vì là một phần mềm rất lâu đời nên nó có hộ trợ làm game trên mọi nền tảng PC, Mobile, Web, Console. Ngôn ngữ lập trình của GameMaker là GameMaker Language (GML) có sự tương đồng với ngôn ngữ JavaScript và C.



Hình 3 Logo của Game Maker



Hình 4 Màn hình làm việc của GameMaker

### RPG Maker.

RPG Maker là một phần mềm chuyên làm về game 2D Pixel Art xuất phát từ Nhật Bản vào năm 1992 bằng ngôn ngữ JavaScript. Như tên gọi của nó RPG viết tắt của Role-Playing Game hay còn gọi là game nhập vai phưu lưu giả tưởng có cốt truyện một thể loại game đến từ Nhật Bản. Hỗ trợ làm game trên các nền tảng chính PC, Console.

A video game screen with text

Description automatically generated

Hình 5 Logo của RPG Maker

A screenshot of a video game

Description automatically generated

Hình 6 Màn hình làm việc của RPG Maker

### Unreal Engine:

Unreal Engine là một công cụ làm game được phát triển bởi Epic Game viết bằng ngôn ngữ C++, được giới thiệu lần đầu tiên vào năm 1998. Mục đích ban đầu của phần mềm là tạo ra các tựa game bắn súng góc nhìn thứ nhất. Một trong những công cụ cho ra đồ họa cực kỳ chất lượng, mã nguồn mở có thể truy cập được trên GitHub và sử dụng một mô hình tài chính người dùng phải trả tiền cho công cụ khi doanh thu game đạt được hơn 1 triệu đô la ngoài việc làm game Unreal Engine cũng được sử dụng trong phim ảnh ví dụ như The Mandalorian của Disney.



Hình 7 Logo của Unreal Engine

A screenshot of a computer

Description automatically generated

Hình 8 Màn hình làm việc của Unreal Engine

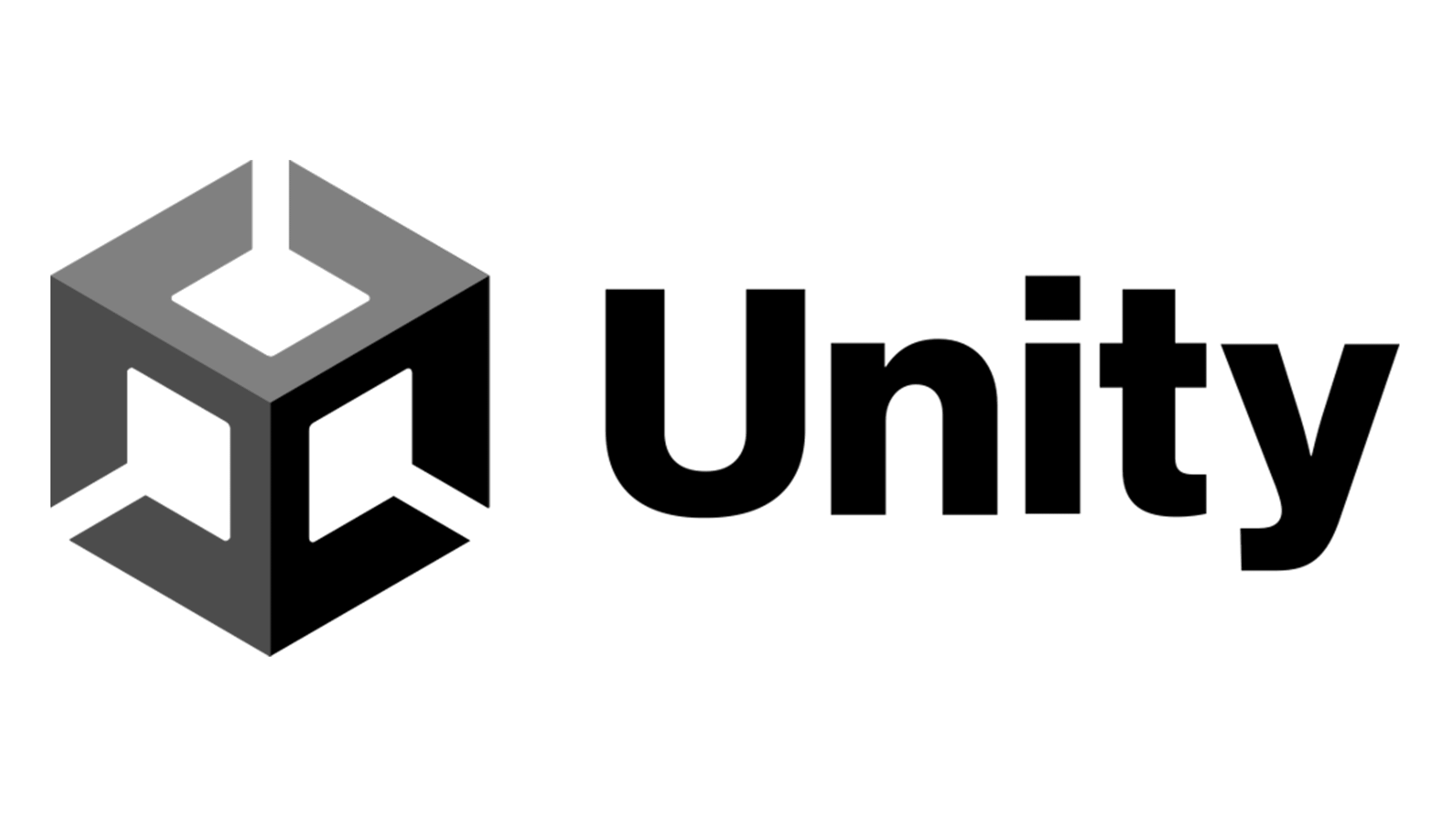
# : CƠ SỞ LÝ THUYẾT

## Unity Engine



### Tổng quát về Unity Engine

Unity Engine được phát triển bởi Unity Technologies, lần đầu được giới thiệu và ra mắt vào năm 2005. Phần mềm hỗ trợ những nền tảng PC, mobile, console và VR (thực tế ảo). Rất phổ biến trong việc phát triển các tựa game trên điện thoại và được coi là một phần mềm rất dễ sự dụng cho người mới bắt đầu, nổi tiếng trong cộng đồng người làm game một người và các nhóm làm game nhỏ lẻ với mô hình kinh doanh thu phí khi đạt được khoảng doanh thu nhất định.



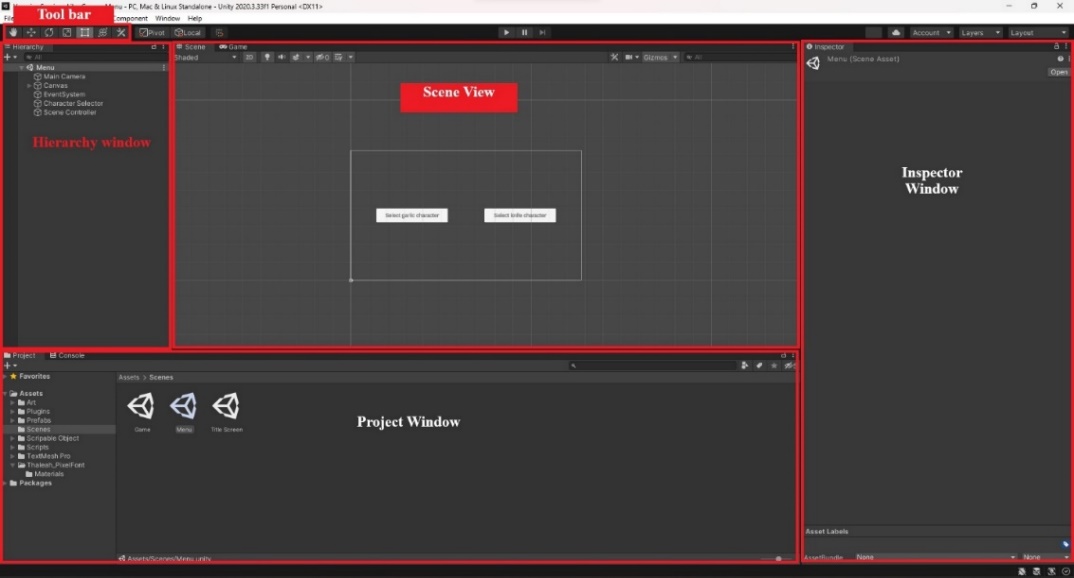
Hình 9 Logo của Unity

### Lý do sử dụng Unity Engine

Các lý do chính cho việc chọn unity là công cụ làm game:

* Unity Engine hoàn toàn miễn phí và chỉ bắt đầu mô hình thu phí khi trò chơi bắt đầu kiếm được doanh thu từ 100 nghìn đô la.
* Giao diện người dùng rất hoàn thiện, dễ dàng sử dụng.
* Các bài học làm những game cơ bản để bắt đầu làm quen với các tính năng chính và bắt đầu các ý tưởng ban đầu để xây dựng một tựa game qua Unity Learn.
* Dễ dàng cài đặt và tích hợp với Visual Studio để viết code.
* Cộng đồng sử dụng Unity rất lớn nên dễ dàng tìm kiếm và đặt những câu hỏi về các vấn đề khúc mắc.
* Cửa hàng asset đa dạng.

### Màn hình làm việc chính của Unity và các thuộc tính cơ bản của Unity



Hình 10 Màn hình làm việc chính của Unity

Màn hình làm việc chính của Unity Engine bao gồm những thành phần chính:

* Hierarchy Window
* Tool Bar
* Inspector Window
* Project Window
* Scene View và Game View

#### Hierarchy Window

Hierarchy Window là nơi sắp xếp mọi GameObject trong project. Nếu ta thêm những GameObjects vào màn hình trò chơi thì nó sẽ được liệt kê tại Hierachy Window. GameObject là một thứ tượng trưng cho mọi thứ trong Unity Engine mà chúng ta muốn hiện thị lên trên màn hình chính như vật thể 2D, 3D, ánh sáng, hiệu ứng, UI,…

A screenshot of a computer

Description automatically generated

Hình 11 Một số GameObject

#### Tool bar

Là tập hợp các công cụ phổ biến thường sử dụng trong quá trình làm việc với Unity như di chuyển một vật thể, phóng to thu nhỏ, xoay, . . .

A screenshot of a video game

Description automatically generated

Hình 12 Di chuyển lớp Player

#### Inspector Window

Inspector Window là nơi tập hợp tất cả các thuộc tính của một GameObject bao gồm những thuộc tính cơ bản là tọa độ. Ở inspector window chúng ta có thể thao tác với các script gắn vào GameObject hay thay đổi các thuộc tính của nó.

A screenshot of a computer

Description automatically generated

Hình 13 Hình ảnh Inspector của lớp Player

#### Project Window

Project Window là nơi chúng ta có thể tìm tất cả các tập tin có trong project. Project Window có chức năng tương tự một cửa sổ tệp tin, chúng ta có thể thao tác với nó như tạo các folder, kéo thả các file từ bên ngoài vào trong các folder tương tự cũng có thể kéo thả các assets từ trong project window vào màn hình thao tác chính của project.

A screenshot of a computer

Description automatically generated

Hình 14 Cấu trúc của folder trong Project Window

#### Scene View và Game View

Trung tâm của giao diện Unity cơ bản là Scene View. Đây là màn hình mà chúng ta sẽ thao tác để tạo ra màn hình chính của game và cũng như là góc nhìn của người chơi. Ở trong scene view thì chúng ta sẽ xem được mọi góc, thao tác với các vật thể ở trên đó và xem chúng ở mọi góc độ. Game View là màn hình của người chơi hoặc sẽ chuyển sang khi chúng ta bắt đầu Play Mode để chơi thử.

A screen shot of a video game

Description automatically generated

Hình 15 Scene View bao gồm GameObject Player và các thuộc tính

A screenshot of a video game

Description automatically generated

Hình 16 Game View màn hình trò chơi cuối cùng mà người chơi sẽ trải nghiệm

### Các khái niệm trong Unity

#### Asset

Là một kho lưu trữ tài nguyên trong một project của Unity. Những tài nguyên này có thể là hình ảnh, âm thanh, mô hình 3D, các hiệu ứng, . . . để dùng trong cho trò chơi. Vì vậy nên mọi project của unity đều lưu trữ trong thư mục tên Asset

A screenshot of a computer menu

Description automatically generated

Hình 17 Cấu trúc một folder asset

#### Scene

Scene trong Unity có thể được xem là những màn hình riêng lẻ phục vụ những mục đích khác nhau như màn hình chính khi vào một trò chơi, màn hình lựa chọn hoặc là những màn chơi khác nhau, viêc có những scene giúp cho chúng ta có thể dễ dàng hơn kiểm soát những màn chơi và các màn hình khác với nhau thay vì làm tất cả trên một Scene.

A screenshot of a computer

Description automatically generated

Hình 18 Ví dụ về một folder Scenes

#### GameObject

Là lớp thuộc tính cơ bản nhất của Unity tượng chưng cho các vật thể nhân vật, đồ vật, . . . trên màn hình chính.

A screenshot of a computer

Description automatically generated

Hình 19 Ví dụ về một GameObject

#### Script

Script là một thành phần quan trọng của Unity có thể xem như việc vận hành Unity phải hầu hết thông qua các script. Chúng ta có thể viết code dưới dạng C# hoặc những ngôn ngữ khác. Nhưng đa số người dùng Unity sử dụng ngôn ngữ chính là C# do hiệu năng nó mang lại. Ngoài C# chúng ta sẽ phải dùng thêm một số API riêng của Unity để thực hiện những chức năng nhất định. Để một script có thể hoạt động thì chúng ta phải gắn chúng một một đối tượng nhất định trên màn chơi.

A screenshot of a computer

Description automatically generated

Hình 20 Ví dụ về một đối tượng được gắn Script

#### Transform

Là lớp biểu hiện vị trí của 1 GameObject trên màn hình, nó chứa các thông số như Position (vị trí), Rotation (xoay), Scale (tỷ lệ).

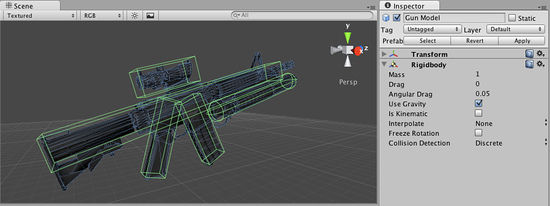
A screenshot of a computer

Description automatically generated

Hình 21 Transform của lớp Player trên màn hình

#### Collider

Là một vòng bao tàng hình với các hình dạng khác nhau với tác dụng chính bao ngoài một vật thể và giúp tạo ra những tương tác thông qua hàm của chúng.



Hình 22 Ví dụ về một collider

#### Rigidbody

Rigidbody là thành phần giúp giả lập vật lý cho một vật thể giúp chúng chân thật hơn khi tác động lẫn nhau

A screenshot of a computer

Description automatically generated

Hình 23 Lớp Rigidbody2D khi thêm vào GameObject Player

#### Animation

Là công cụ giúp chúng ta có thể thực hiện hoạt ảnh giúp game chúng ta chân thực hơn. Trong menu animation chúng ta có thêm một khái niệm nữa là state machine. State Machine là một định dạng những trạng thái mà một hoạt ảnh sẽ thực hiện và các điều kiện để thực hiện các hoạt ảnh đó.

A screenshot of a computer

Description automatically generated

Hình 24 Animator di chuyển của người chơi

#### Unity UI

Là một tổ hợp các công cụ tạo lên một UI cho người dùng gồm các thành phần như button, text, canvas, . . . Có thể truy cập các thuộc tính bằng Unity.UI.

A screenshot of a video game

Description automatically generated

Hình 25 Hình ảnh UI của game

#### Scripable Object

Là một dạng cấu trúc dữ liệu của Unity với tác dụng chính là để lưu trữ các dữ liệu ở những lần chơi và sử dụng để thay thế các dữ liệu hay chỉ số không thay đổi trong một trò chơi để giảm bộ nhớ sử dụng ở những lần gọi hay cũng có thể sử dụng như một cơ sở dữ liệu. Cách sử dụng là gọi ScripableObject ở phần public class của một script và tạo menu asset để truy cập tạo các object trên.

A screen shot of a computer program

Description automatically generated

Hình 26 Ví dụ về một script sử dụng scripable object

A screenshot of a computer

Description automatically generated

Hình 27 Scriptable Object tạo từ editor

#### MonoBehaviour

MonoBehaviour là một lớp cơ bản và mặc định của mọi script được tạo ra từ Unity Engine, luôn được tạo với một template nhất định bao gồm lớp Start() và lớp Update(). MonoBehaviour đảm nhiệm vai trò là bộ máy vận hành của Unity khi thực hiện các câu lệnh chạy khi game bắt đầu khởi tạo và khi game đang chạy.

A computer screen shot of a program

Description automatically generated

Hình 28 Ví dụ về một script MonoBehavior khi được khởi tạo

## Các công cụ hỗ trợ khác

### Visual Studio

Là một chương trình biên dịch code đến từ Microsoft bao gồm tất cả các chức năng chính như viết code, build, debug code, test, version control, collaborate, deploy. Với số lượng lớn ngôn ngữ tích hợp bao gồm C, C++, .NET, C#,… và các ngôn ngữ khác như python, ruby, node js dưới dạng các plugin hỗ trợ bên ngoài trên hết là hỗ trợ cả Unity Engine.

### GIMP

Là một chương trình chỉnh sửa hình ảnh miễn phí mã nguồn mở có thể truy cập trên github, dù không có các thế mạnh như các công cụ khác vì bản chất tối giản nhưng gimp cũng thực hiện được các chức năng chỉnh sửa ảnh, vẽ, chuyển đổi các định dạng hình ảnh nên bản chất cũng không kém cạnh các sản phẩm khác là bao và trên hết phần mềm hoàn toàn miễn phí dễ dàng cài đặt thông qua microsoft store ở trên window.

### Github

Github là một dịch vụ cung cấp kho lưu trữ code trên web cho các dự án phần mềm và cũng như theo dõi sự thay đổi, quản lý code. Github cung cấp dịch vụ bao gồm miễn phí và trả phí. Với hơn 100 triệu người dùng khẳng định được vị thế của bản thân trong giới công nghệ là một trong những công cụ lưu trữ, quản lý code hàng đầu thế giới và dễ dàng sử dụng thông qua phiên bản Github Desktop khiến việc thao tác không cần phải sử dụng lệnh thông qua command prompt.

# : LÊN Ý TƯỞNG VÀ THIẾT KẾ GAME



## Lên ý tưởng

Trong thể loại game sinh tồn 2D hiện nay nổi lên một tựa game rất nổi tiếng là Vampire Survivor với lối chơi rất đơn giản và không hề phức tạp, mục tiêu chính của người chơi là sống sót và tiêu diệt con trùm của bản đồ đó để hoàn thành màn chơi. Trò chơi mang đến hệ thống vũ khí và nội tại, lên cấp khi tiêu diệt kẻ thù, tiêu diệt quái lớn sẽ rơi các rương vật phẩm ngẫu nhiên và tiền để nâng cấp các kỹ năng ngoài màn chơi để tăng sức mạnh ở các lần chơi tiếp theo. Vì vậy bản thân trò chơi có giá trị chơi lại rất cao vì sở hữu số lượng lớn các vũ khí thú vị, các con quái trông có vẻ dễ dàng tiêu diệt nhưng lại rất thử thách khả năng sinh tồn của người chơi vì càng sinh tồn lâu số lượng chúng càng tăng lên.

## Thiết kế game

### Bản đồ

A screenshot of a video game

Description automatically generated

Hình 29 Hình ảnh cấu trúc của bản đồ

Bản đồ trong game được cấu tạo từ 1 tấm pixel với chiều dài và rộng là 20x20

### Nhân vật chính

Nhân vật chính của trò chơi của chúng ta sẽ khởi đầu với 1 trong 2 vũ khí được cho sẵn, nhân vật chính được thiết kế bao gồm có vũ khí khởi đầu, tổng máu tối đa, hồi máu trên giây, tốc độ di chuyển, hệ số sát thương và bán kính hút vật phẩm.

A screenshot of a game

Description automatically generated

Hình 30 Sprite nhân vật chính

### Quái vật

Quái vật trong game được thiết kế với ý tưởng đơn giản sẽ tự khóa mục tiêu vào người chơi sau khi được sinh ra và liên tục di chuyển tới người chơi, gây sát thương khi chạm vào người chơi, mỗi một quái thì lại có chỉ số khác nhau. Khi bị tiêu diệt quái sẽ rơi ra kim cương kinh nghiệm để giúp người chơi lên cấp.

A screenshot of a video game

Description automatically generated

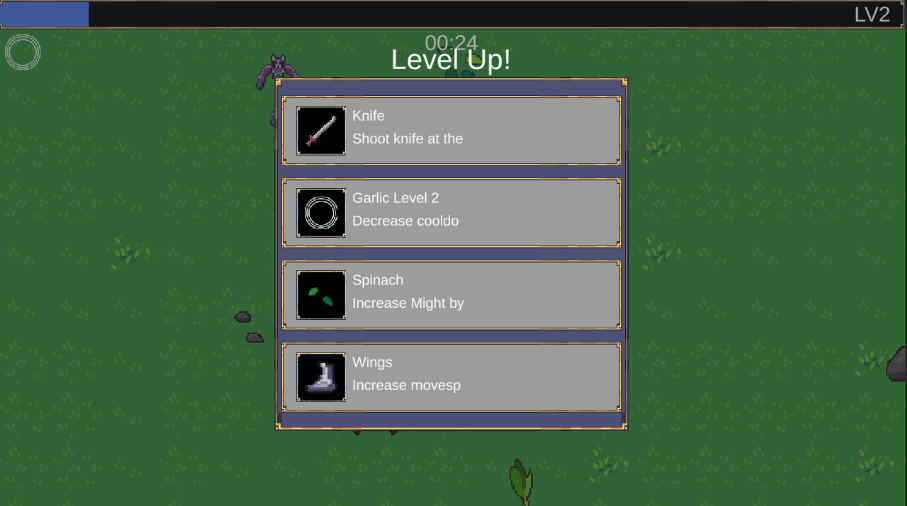
Hình 31 Hình ảnh quái vật Dơi

### Hệ thống vũ khí

Hệ thống vũ khí trong game hoàn toàn tự động và không cần người chơi phải thao tác bất cứ gì ngoài di chuyển nhân vật. Ý tưởng của hệ thống vũ khí bao gồm 3 loại: vũ khí cận chiến, vũ khí tầm xa và nội tại.

### Hệ thống cấp độ và nâng cấp

Hệ thống cấp độ sẽ giúp tăng tiến sức mạnh của người chơi trong quá trình chơi game, khi lên cấp người chơi sẽ được lựa chọn một trong số những vũ khí hoặc nội tại chưa sở hữu hoặc nâng cấp các vật phẩm hiện có của bản thân. Khi đang trong tình trạng lên cấp toàn bộ trò chơi sẽ ngưng lại cho đến khi người chơi chọn cho mình nâng cấp.



Hình 32 Màn hình lên cấp và hệ thống nâng cấp

### Điều khiển

Điều khiển nhân vật cơ bản di chuyển với 4 hướng thông qua các nút w, a, s, d. Các vũ khí tầm xa sẽ tự động bắn trong 1 khoảng thời gian nhất định tại hướng mà người chơi đang di chuyển.

### Các hệ thống khác

Ngoài các hệ thống trên còn các hệ thống khác cho phép người chơi dừng lại trong quá trình chơi game, thống kê các chỉ số và vật phẩm hiện có của nhân vật, thoát game ra màn hình chính, bắt đầu một lần chơi mới, hệ thống tự sinh quái, tự hồi máu, hiển thị sát thương khi đánh quái và các hiệu ứng khi nhận sát thương của người chơi và quái vật.

# : PHÂN TÍCH HỆ THỐNG

## Phân tích các actor chính

Các actor chính của game bao gồm:

* Người chơi (player): là nhân vật chính do người chơi điều khiển thực hiện các thao tác với môi trường của game và với quái vật
* Quái vật: là các nhân vật có AI đơn giản điều khiển di chuyển luôn luôn hướng tới người chơi và gây sát thương cho người chơi.

## Xác định usercase và đặc tả

### Xác định usercase

Hệ thống có các usercase sau:

* Di chuyển của người chơi: Di chuyển bằng 4 hướng theo các phím w, a, s, d.
* AI của quái vật: Luôn luôn di chuyển đến vị trí người chơi.
* Hệ thống vũ khí và nội tại: Người chơi có một hệ thống vũ khí có chức năng khác nhau và các nội tại để gia tăng các chỉ số.
* Lên cấp: Lên cấp khi thu thập đủ lượng kinh nghiệm từ đá quý.
* Nâng cấp: Khi người chơi lên cấp sẽ xuất hiện bảng nâng cấp các vũ khi, nội tại những vũ khí và nội tại chưa sở hữu sẽ được thêm vào hành trang sau khi lựa chọn.
* Xử lý va chạm của người chơi: Người chơi sẽ bị gây sát thương khi va chạm phải quái vật
* Xử lý va chạm của quái vật: Quái vật sẽ bị gây sát thương khi chạm phải vũ khí từ người chơi.

### Đặc tả usercase

Bảng 1 Sơ đồ usercase di chuyển của người chơi

|  |  |
| --- | --- |
| **Mô tả** | Di chuyển của người chơi |
| **Tác nhân** | Người chơi |
| **Tiền điều kiện** | Màn chơi bắt đầu |
| **Hậu điều kiện** | Nhân vật bắt đầu di chuyển theo các hướng ấn ở bàn phím |
| **Luồng sự kiện** | 1. Người chơi ấn các phím w, a, s, d để di chuyển nhân vật. 2. Xử lý di chuyển của nhân vật 3. Hệ thống bắt đầu thực hiện hoạt ảnh di chuyển của nhân vật. |
| **Luồng ngoại lệ** | 2a. Hệ thống sẽ thay đổi hoạt ảnh tương ứng với việc đứng im và di chuyển.  2b. Khi người chơi chết thì sẽ không thể di chuyển được nữa. |

Bảng 2 Usercase di chuyển của quái vật

|  |  |
| --- | --- |
| **Mô tả** | AI của quái vật |
| **Tác nhân** | Quái vật |
| **Tiền điều kiện** | Khi quái vật được sinh ra |
| **Hậu điều kiện** | Quái vật di chuyển đến vị trí của người chơi |
| **Luồng sự kiện** | 1. Quái vật được sinh ra. 2. Xử lý di chuyển của quái vật. 3. Hệ thống bắt đầu thực hiện hoạt ảnh di chuyển của nhân vật. |
| **Luồng ngoại lệ** | Không có |

Bảng 3 Usercase hệ thống vũ khí và nội tại

|  |  |
| --- | --- |
| **Mô tả** | Hệ thống vũ khí và nội tại |
| **Tác nhân** | Người chơi |
| **Tiền điều kiện** | Khi vũ khí và nội tại được người chơi sở hữu. |
| **Hậu điều kiện** | Thực hiện các chức năng của vũ khí và nội tại. |
| **Luồng sự kiện** | 1. Khi vũ khí và nội tại được sở hữu bởi người chơi. 2. Hệ thống thực hiện các chức năng của vũ khí và nội tại. |
| **Luồng ngoại lệ** | 2a. Nếu vũ khí được sở hữu bởi người chơi sẽ thực hiện các chức năng của các vũ khí đó.  2b. Nếu nội tại được sở hữu bởi người chơi sẽ thay đổi các chỉ số của người chơi. |

Bảng 4 Usercase lên cấp

|  |  |
| --- | --- |
| **Mô tả** | Lên cấp |
| **Tác nhân** | Người chơi |
| **Tiền điều kiện** | Thu thập các viên đá quý kinh nghiệm |
| **Hậu điều kiện** | Lên cấp |
| **Luồng sự kiện** | 1. Thu thập đá quý nhận kinh nghiệm. 2. Hệ thống kiểm tra lượng kinh nghiệm nhận được. 3. Lên cấp. |
| **Luồng ngoại lệ** | Không có |

Bảng 5 Usercase nâng cấp

|  |  |
| --- | --- |
| **Mô tả** | Nâng cấp |
| **Tác nhân** | Người chơi |
| **Tiền điều kiện** | Lên cấp |
| **Hậu điều kiện** | Chọn nâng cấp |
| **Luồng sự kiện** | 1. Người chơi lên cấp. 2. Bảng nâng cấp xuất hiện. 3. Người chơi chọn nâng cấp. |
| **Luồng ngoại lệ** | 2a. Có 4 lựa chọn mà người chơi có thể chọn. Hiển thị gồm tên của nâng cấp và mô tả nâng cấp.  3a. Nâng cấp chưa được sở hữu sẽ được thêm vào hành trang còn nâng cấp đã sở hữu sẽ được nâng lên cấp độ tiếp theo. |

Bảng 6 Usercase xử lý va chạm của người chơi

|  |  |
| --- | --- |
| **Mô tả** | Xử lý va chạm của người chơi |
| **Tác nhân** | Người chơi |
| **Tiền điều kiện** | Khi quái vật, vật phẩm chạm vào người chơi |
| **Hậu điều kiện** | Thực hiện các tương tác giữa người chơi và các vật thể va chạm |
| **Luồng sự kiện** | 1. Người chơi chạm phải vật thể. 2. Hệ thống kiểm tra các tương tác giữa người chơi và vật thể va chạm. 3. Thực hiện các tương tác với nhau. |
| **Luồng ngoại lệ** | 1a. Người chơi có thể va chạm được với quái vật, vật phẩm.  3a. Khi người chơi va chạm phải quái vật sẽ bị trừ máu, khi máu tụt về 0 thì màn chơi kết thúc.  3b. Sẽ có một khoảng dừng nhỏ giữa các lần bị khi người chơi bị gây sát thương.  3c. Người chơi nhân được kinh nghiệm khi tiếp xúc với đá quý kinh nghiệm.  4d. Người chơi hồi lượng máu hiện tại khi tiếp xúc với bình máu. |

Bảng 7 Usercase xử lý va chạm của quái vật

|  |  |
| --- | --- |
| **Mô tả** | Xử lý va chạm của quái vật |
| **Tác nhân** | Quái vật, vũ khí |
| **Tiền điều kiện** | Quái vật chạm phải các vũ khí |
| **Hậu điều kiện** | Quái vật bị gây sát thương |
| **Luồng sự kiện** | 1. Quái vật chạm phải vũ khí. 2. Hệ thống kiểm tra sát thương của vũ khí chạm phải quái vật. 3. Quái vật bị trừ máu. |
| **Luồng ngoại lệ** | 3a. Quái vật sẽ có một khoảng thời gian ngắn không nhận sát thương khi vừa bị gây sát thương.  3b. Nếu quái vật bị trừ hết máu thì sẽ chết và rơi ra đá quý kinh nghiệm. |

## Biểu đồ chức năng

### Biểu đồ usercase tổng quát

A diagram of a person with text

Description automatically generated

Hình 33 Biểu đồ usercase tổng quát

### Chức năng di chuyển

A diagram of a person with a diagram

Description automatically generated

Hình 34 Biểu đồ usercase di chuyển

A screenshot of a diagram

Description automatically generated

Hình 35 Biểu đồ hoạt động di chuyển

A diagram of a diagram

Description automatically generated

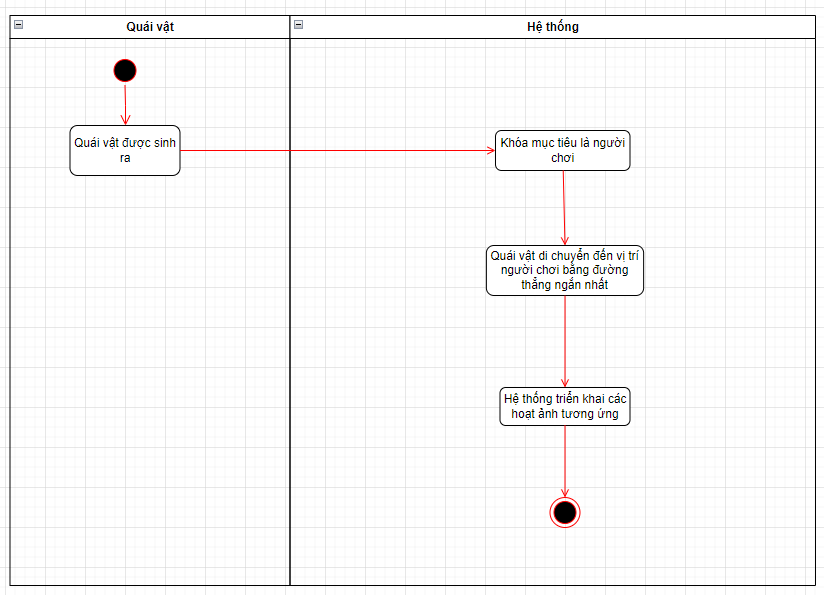
Hình 36 Biểu đồ trình tự di chuyển

### Chức năng AI của quái vật

A diagram of a person with a figure and a person with a person with a person with a person with a person with a person with a person with a person with a person with a person with

Description automatically generated

Hình 37 Biểu đồ usercase AI của quái vật



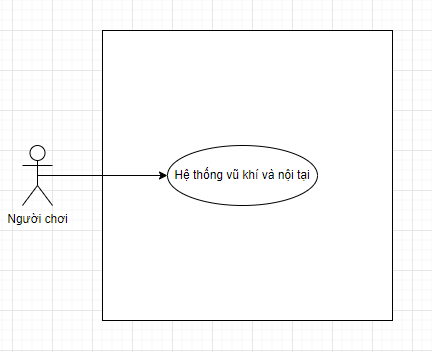
Hình 38 Biểu đồ hoạt động AI của quái vật

A diagram of a diagram

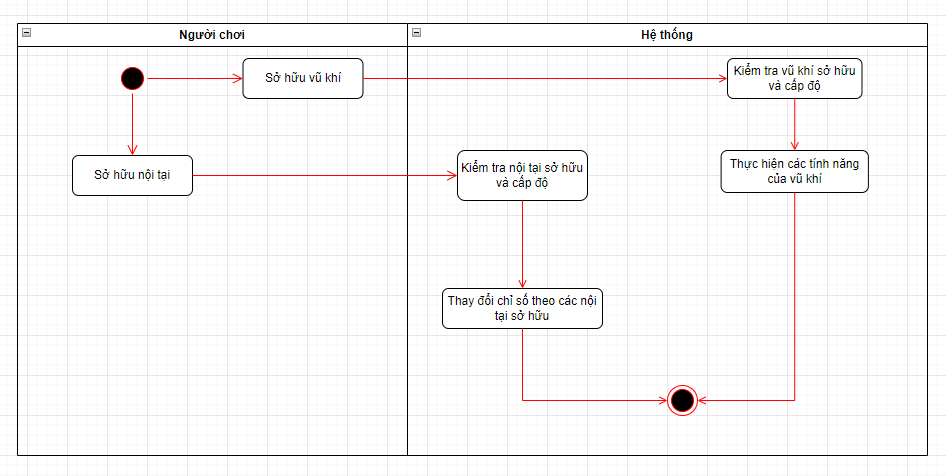
Description automatically generated

Hình 39 Biểu đồ trình tự AI của quái vật

### Chức năng hệ thống vũ khí



Hình 40 Biểu đồ usercase hệ thống vũ khí



Hình 41 Biểu đồ hoạt động của hệ thống vũ khí

A diagram of a diagram

Description automatically generated with medium confidence

Hình 42 Biểu đồ trình tự hệ thống vũ khí

### Chức năng lên cấp

A diagram of a person with a person in the middle

Description automatically generated

Hình 43 Biểu đồ usercase lên cấp

A diagram of a diagram

Description automatically generated with medium confidence

Hình 44 Biểu đồ hoạt động lên cấp

A graph with text and a line

Description automatically generated with medium confidence

Hình 45 Biểu đồ trình tự lên cấp

### Chức năng nâng cấp

A diagram of a person with a stick figure

Description automatically generated

Hình 46 Biểu đồ usercase nâng cấp

A diagram with red lines and black dots

Description automatically generated

Hình 47 Biểu đồ hoạt động nâng cấp

A screenshot of a graph

Description automatically generated

Hình 48 Biểu đồ trình tự nâng cấp

### Chức năng xử lý va chạm của người chơi

A diagram of a person with a figure

Description automatically generated

Hình 49 Biểu đồ usercase xử lý va chạm người chơi

A diagram of a flowchart

Description automatically generated

Hình 50 Biểu đồ hoạt động xử lý va chạm của người chơi

A screenshot of a graph

Description automatically generated

Hình 51 Biểu đồ trình tự xử lý va chạm của người chơi

### Chức năng xử lý va chạm của quái vật

A diagram of a person with a diagram

Description automatically generated

Hình 52 Biểu đồ usercase xử lý va chạm của quái vật

A screenshot of a diagram

Description automatically generated

Hình 53 Biểu đồ hoạt động xử lý va chạm của quái vật

A grid with lines and words

Description automatically generated with medium confidence

Hình 54 Biểu đồ trình tự xử lý va chạm của quái vật

# XÂY DỰNG GAME VÀ DEMO

## Xây dựng game

Link demo: <https://github.com/Duk1321/DATN_1951060614_DuongGiapDuc>

## Demo

A screen shot of a video game

Description automatically generated

Hình 55 Màn hình bắt đầu game

A screenshot of a computer

Description automatically generated

Hình 56 Menu chọn nhân vật

A screenshot of a video game

Description automatically generated

Hình 57 Màn hình chính trong game

A screenshot of a video game

Description automatically generated

Hình 58 Màn hình dừng game

A screenshot of a video game

Description automatically generated

Hình 59 Màn hình lên cấp và nâng cấp

A screenshot of a video game

Description automatically generated

Hình 60 Màn hình kết quả màn chơi

# KẾT LUẬN

## Kết luận

Qua việc thực hiện báo cáo này và làm một bản demo em đã được tiếp xúc một phần gần hơn với việc làm game, học cách viết những dòng lệnh unity đầu tiên, làm quen với những vật thể trong game, học những phương pháp tiếp cận để giải quyết bài toán được đặt ra. Và em cũng nhận ra rằng việc làm game là một việc đòi hỏi rất cao về khả năng sáng tạo, kiên trì. Dù chỉ những chức năng kể trên đã hoạt động theo mô tả nhưng khả năng tái sử dụng khá thấp và em nghĩ em cần thêm nhiều thời gian hơn để có thể thực sự tìm ra một lối đi rõ ràng hơn, qua đó cũng thấy rằng để hoàn thành một game hoàn chỉnh cũng cần phải có một đội ngũ những người đảm nhận những vai trò riêng trong quy trình làm một tựa game để mang đến một sản phẩm thực thụ và chỉnh chu đến người chơi.

## Định hướng phát triển

Dù demo đã hoàn thiện nhưng em cũng đã bóc tách ra được những thứ sau đây để khiến game hoàn chỉnh hơn:

* Tạo những con quái vật mới để trò chơi thêm đa dạng.
* Thêm những nâng cấp và nội tại để khiến game có giá trị chơi lại nhiều hơn.
* Thiết kế thêm những màn chơi mới, bản đồ mới.
* Thêm hiệu ứng âm thanh cho trỏ chơi.
* Chạy tựa game trên hệ điều hành điện thoại.
* Hoàn thiện hệ thống vũ khí và nội tại để dễ dàng hơn cho việc phát triển thêm những vũ khí và nội tại mới.
* Hoàn thiện giao diện người dùng.

# TÀI LIỆU THAM KHẢO

[1] <https://docs.unity3d.com/Manual/index.html>

[2] <https://learn.unity.com/>

[3] <https://vampire-survivors.fandom.com/wiki/Vampire_Survivors_Wiki>

[4] <https://www.history.com/topics/inventions/history-of-video-games>

[5] <https://builtin.com/gaming>

# CÁC ASSET SỬ DỤNG TRONG ĐỀ TÀI

[1] <https://seliel-the-shaper.itch.io/character-base>

[2] <https://opengameart.org/content/bat-sprite>

[3] <https://opengameart.org/content/lpc-terrain-repack>

[4] <https://opengameart.org/content/admurins-armory>

[5] <https://opengameart.org/content/admurins-potions>

[6] <https://opengameart.org/content/zelda-like-tilesets-and-sprites>

[7] <https://opengameart.org/content/gem-icons>