

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

KHOA ĐIỆN TỬ

Bộ môn: Công Nghệ Thông Tin



BÀI TẬP LỚN

MÔN HỌC

LẬP TRÌNH GAME 3D VỚI UNITY

Sinh viên : Phạm Sỹ Quang

Lớp : 55KMT

Giáo viên hướng dẫn : ThS. Đỗ Duy Cốp

Thái Nguyên – 2023

BÀI TẬP LỚN

MÔN HỌC: Lập trình game 3D với Unity

BỘ MÔN : Công nghệ thông tin

Sinh viên: *Phạm Sỹ Quang*

Lớp: *55KMT.01*

Ngành: Tin học công nghiệp

Giáo viên hướng dẫn: ThS. Đỗ Duy Cốp

Ngày giao đề: 19/05/2023

Ngày hoàn thành: 05/06/2023

1. Tên đề tài: Lập trình game 2D Frustration Frenzy

2. Nội dung thuyết minh tính toán:

Sử dụng những kiến thức đã học và trong quá trình tự tìm hiểu để xây dựng một game 2D cơ bản.

Tạo giao diện tương tác với người dùng, cho phép người dùng có thể điều khiển nhân vật, chạy, nhảy, vượt chướng ngại vật,...

Tạo ra nhiều màn khác nhau, tăng độ khó mỗi khi người chơi vượt qua màn và dành chiến thắng.

Có hiệu ứng âm thanh, hình ảnh tăng độ tương tác cho người chơi.

Xuất bản thành một ứng dụng cụ thể và có thể trải nghiệm trên thiết bị khác.

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

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

GIÁO VIÊN HƯỚNG DẪN

(Ký ghi rõ họ tên)

NHẬN XÉT CỦA GIÁO VIÊN CHẤM

.....

.....

.....

.....

.....

.....

.....

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

GIÁO VIÊN CHẤM

(Ký ghi rõ họ tên)

MỤC LỤC

MỤC LỤC	4
DANH MỤC HÌNH ẢNH	6
LỜI NÓI ĐẦU	1
CHƯƠNG 1. GIỚI THIỆU CHUNG VỀ GAME	2
1.1. Giới thiệu game	2
1.2. Mô tả quá trình chơi game	3
1.3. Yêu cầu với game.....	3
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....	4
2.1. Tổng quan về Engine Unity	4
2.1.1. <i>Unity là gì?</i>	4
2.1.2. <i>Sơ lược hình thành và phát triển của Unity</i>	6
2.1.3. <i>Tính năng của Engine Unity</i>	6
2.2. Tổng quan về Game Engine	7
2.2.1. <i>Khái niệm</i>	7
2.2.2. <i>Mục đích</i>	7
2.3. Tổng quan kiến trúc Game Engine	8
CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ	10
3.1. Các chức năng của hệ thống.....	10
3.2. Biểu đồ hoạt động	10
3.3. Biểu đồ tuần tự	11
CHƯƠNG 4. XÂY DỰNG CHƯƠNG TRÌNH	12
4.1. Xây dựng giao diện	12
4.2. Code chương trình.....	15
KẾT LUẬN	17
TÀI LIỆU THAM KHẢO	18

DANH MỤC TỪ VIẾT TẮT

STT	Từ viết tắt	Từ đầy đủ
1	GUI	Graphical Use Interface

DANH MỤC HÌNH ẢNH

Hình 1. Giao diện Backgroud game	2
Hình 2. Giao diện nhân vật ban đầu game	3
Hình 3. Hình ảnh Logo của Unity.....	4
Hình 4. Hình ảnh khi build ứng dụng game ra nhiều nền tảng khác nhau	5
Hình 5. Tổng quan kiến trúc UnityAPI.....	8
Hình 6. Biểu đồ hoạt động khi bắt đầu game	10
Hình 7. Biểu đồ điều khiển nhân vật.....	11
Hình 8. Biểu đồ tuần tự khi bắt đầu chơi game	11
Hình 9. Hình ảnh giao diện ban đầu khi vào Game	12
Hình 10. Đoạn code khi người dùng nhấn nút START	12
Hình 11. Đoạn code nói về việc điều khiển nhân vật di chuyển	13
Hình 12. Đoạn code script sử dụng trong Game.....	13
Hình 13. Giao diện khi người chơi ở vạch đích màn 1	13
Hình 14. Giao diện ban đầu khi ở màn 2.....	14
Hình 15. Hình ảnh giao diện màn 2.....	14
Hình 16. Hình ảnh kết thúc Game	14
Hình 17. Code chương trình lấy điểm di chuyển của vật trong Game	15
Hình 18. Kiểm tra trạng thái của người chơi	15
Hình 19. Code chuyển màn hình tiếp theo khi người chơi giành chiến thắng	16

LỜI NÓI ĐẦU

Ngày nay, xã hội ngày càng phát triển mạnh mẽ, nhiều ngành nghề được xây dựng và phát triển không ngừng. Trong đó, nhiều ngành nghề đem lại lợi ích về mặt vật chất, của cải phục vụ con người, nhưng cũng có nhiều ngành nghề lại đem lại lợi ích về mặt tinh thần, giúp con người có thể giải tỏa căng thẳng sau khoảng thời gian làm việc, lao động mệt mỏi. Một trong những ngành nghề đó là “Lập trình Game”, một ngành nghề tương đối mới và đang được phát triển mạnh mẽ vào những năm gần đây.

Ở nhiều quốc gia phát triển, ngành công nghệ giải trí nói chung và ngành lập trình Game nói riêng đều có những thành tựu lớn và được nhiều sự coi trọng của nhà nước. Đối với nước ta, những năm gần đây cũng đã và đang có những game “nhanh chóng nổi rộ” nhưng cũng “nhanh chóng lụi tàn”.

Nhìn chung, để có thể xây dựng một ứng dụng Game không phải là một công việc quá khó, không cần đến mức chuyên nghiệp cũng có thể làm được một Game đáp ứng được những tiêu chuẩn chung. Nhưng, để có thể tạo ra những “tác phẩm” lớn mang dấu ấn của nước ta thì lại không phải là một câu chuyện dễ dàng. Có thể nói, đây đang là một vấn đề nan giải vì để có thể tạo ra một game hay và được nhiều người chơi thì còn phụ thuộc vào nhiều yếu tố như: ý tưởng, hình ảnh, âm thanh, ...

Chính vì vậy, trong đề tài này em xin được áp dụng những kiến thức đã học trên lớp cũng như trong quá trình tự tìm hiểu của mình để xây dựng game có tên là Frustration Frenzy, đây là một tựa game kết hợp với yếu tố phiêu lưu, giải trí cùng sự khéo léo trong giao diện với góc nhìn của người chơi là ở góc nhìn thứ 3 (Góc nhìn ngang).

Em cũng xin chân thành cảm ơn thầy Đỗ Duy Cốp đã cung cấp cho em những kiến thức về môn Lập trình Game để em có thể hoàn thành bài này. Em nhận thấy bài của mình cũng còn nhiều những thiếu sót cũng như còn nhiều vấn đề chưa nói rõ được nên em cũng rất mong nhận được những ý kiến, đóng góp từ phía thầy cô và những người đọc.

Em xin chân thành cảm ơn!

CHƯƠNG 1. GIỚI THIỆU CHUNG VỀ GAME

1.1. Giới thiệu game

a) Cốt truyện game

Nội dung game là một hành trình của RoboCop, trong quá trình phiêu lưu để tìm kiếm kho báu, cậu đã vô tình lạc vào trong một hang động dưới lòng đất. Game sẽ là quá trình RoboCop tìm kiếm kho báu, tránh đi những cạm bẫy và tìm đường quay trở lại mặt đất.

Cuộc hành trình đầy khó khăn của RoboCop sẽ như thế nào?

Những khó khăn, những cạm bẫy đầy chết chóc sẽ như thế nào?

Hãy giúp cho RoboCop cùng vượt qua hang động và tiến đến kho báu cho chính mình.

b) Đồ họa giao diện game

Đồ họa giao diện game được thiết kế với phong màu cơ bản, tạo cho người chơi có thể có cảm giác thoải mái, tránh nhiều chi tiết đang xen gây cảm giác khó chịu cho người chơi.



Hình 1. Giao diện Background game



Hình 2. Giao diện nhân vật ban đầu game

c) Âm thanh game

Cùng với giao diện đồ họa game, cũng phải nói đến chất lượng âm thanh. Những âm thanh thay đổi khi nhân vật nhảy, qua màn, chết, ... đều có những âm thanh riêng biệt để người chơi có những trải nghiệm riêng biệt.

1.2. Mô tả quá trình chơi game

Game khi bắt đầu. Người dùng cần bấm nút “START” để có thể vào giao diện chơi, người chơi sử dụng bàn phím để điều khiển nhân vật qua phía bên trái, phải, ngẩng lên trên, khi người chơi di chuyển thì Camera sẽ tự động di chuyển theo người chơi. Người chơi điều khiển nhân vật để có thể tránh chạm bẫy và tiến về đích, trong quá trình tiến về đích, nếu người chơi vô tình va chạm vào nhưng chướng ngại thì sẽ chết và quay về điểm hồi sinh ban đầu.

1.3. Yêu cầu với game

- Game cần phải có dung lượng không quá cao, tốc độ xử lý cần phải nhanh.
- Giao diện dễ nhìn, không gây cảm giác khó chịu cho người chơi.
- Khi người chơi chết, người chơi sẽ được hồi sinh tại điểm xuất phát.
- Âm thanh và hình ảnh phù hợp với nội dung và giao diện của game.
- Phân tích game theo hướng đối tượng cụ thể, rõ ràng.
- ...

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về Engine Unity

2.1.1. Unity là gì?

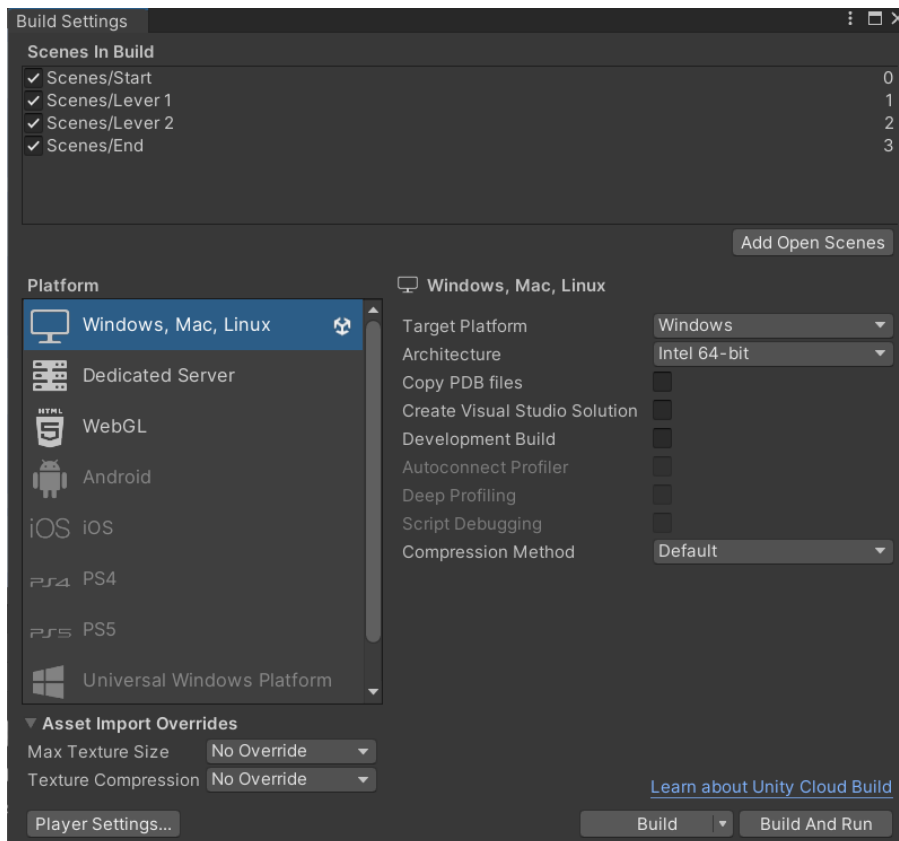


Hình 3. Hình ảnh Logo của Unity

Đã qua một khoảng thời gian làm game trên nền tảng Flash căn bản và buồn chán với những chuyển động cứng nhắc. Unity mang lại một sức mạnh kỳ diệu cho nhân vật mà chúng ta muốn thể hiện sống động hơn trong một không gian 3 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 một cảm giác khác lạ và hào hứng 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 đầy huyền ảo.

Unity được dùng để làm video game, hoặc những nội dung có tính tương tác như thể hiện kiến trúc, hoạt hình 2D, 3D thời gian thực. Unity tương tự như với Director, Blender, Virtools hay Torque Game Builder trong khía cạnh dùng môi trường đồ họa thích hợp ở quá trình phát triển game là chính.

Unity là một trong những Engine được giới làm game không chuyên cực kỳ ưa chuộng bởi khả năng tuyệt vời của nó là phát triển trò chơi đa nền tảng. Ứng dụng có thể chạy được trên nhiều nền tảng khác nhau như: Windows, Mac OS, Android. Game cũng có thể chơi trên trình duyệt web thông qua plugin Unity Web Player.



Hình 4. Hình ảnh khi build ứng dụng game ra nhiều nền tảng khác nhau

Vào năm 2009, unity đã nằm trong top 5 game engine tốt nhất trong việc sản xuất game với chỉ sau 4 năm phát triển. Lượng tài liệu hướng dẫn Unity rất phong phú. Hơn thế nữa nó còn có sẵn một cộng đồng cực lớn với diễn đàn riêng. Bất cứ điều gì không hiểu người lập trình đều có thể thoải mái hỏi và nhận được câu trả lời nhanh chóng, tận tâm.

Sức mạnh: Unity có thể tạo ra được nhiều loại game 2D, 3D đa dạng, dễ sử dụng với người làm game chưa chuyên nghiệp, chất lượng cao, chạy được trên hầu hết các hệ điều hành.

Sự tiện lợi: Nếu chúng ta là một người chuyên dùng 3Dmax hay Maya hoặc phần mềm mã nguồn mở Blender thì quả thật là tuyệt, chúng ta sẽ có một lợi thế lớn khi viết game trên Unity này. Bởi việc tạo các mô hình 2D, 3D sẽ trở lên dễ dàng hơn rất nhiều, việc kết hợp giữa người lập trình và người thiết kế các mô hình nhanh chóng và hiệu quả hơn. Trong Unity chúng ta có thể import trực tiếp các file mô hình đang thiết kế và sẽ thiết kế hoàn thiện tiếp nếu chưa xong trong khi đó công việc chỉ diễn ra một lần. Không như việc phải dùng các công cụ khác để thực hiện viết game chúng ta sẽ phải xuất chúng ra một dạng nào đó và mỗi lần sửa lại phần mô hình chúng ta lại phải import lại, và như thế là quá mất thời gian trong việc tạo và chỉnh sửa các mô hình theo ý muốn. Việc đặt các thuộc tính vật lý trong Unity cũng cực kỳ dễ dàng và hỗ trợ

sẵn nhiều chức năng.

2.1.2. Sơ lược hình thành và phát triển của Unity

a) Lịch sử hình thành

Phần lõi của Unity ban đầu được xây dựng bởi Joachim Ante vào năm 2001. Sau đó được công ty hình thành vào năm 2005 và bắt đầu với phiên bản 1.0. Đến năm 2007, Unity được nâng lên phiên bản 2.0. Unity bắt đầu được hỗ trợ iPhone vào năm 2008. Vào tháng 6/2010, Unity chính thức hỗ trợ Android và cho ra đời phiên bản 3.0 có hỗ trợ Android vào tháng 9/2010 và bây giờ là phiên bản 5. Có thể thấy rằng tốc độ phát triển của Unity khá nhanh.

b) Giải thưởng

Unity đã đạt được nhiều giải thưởng lớn như sau:

Năm 2006, Unity đạt “Best Use of Mac OS Graphics” tại Apple’s WWDC. Đây là lần đầu tiên một công cụ phát triển game đạt được chất lượng do giải thưởng uy tín này đưa ra.

- Năm 2009, Unity Technologies có tên trong “Top 5 công ty game của năm” do Gamasutra tổ chức.
- Năm 2010, Unity đoạt giải Best Engine Finalist do Develop Manazine bình chọn, giải Technology Innovation Award của Wall Street Journal ở thể loại phần mềm.

c) Khách hàng

Unity được trên 250.00 người đăng ký sử dụng gồm Bigpoint, Cartoon Network, Coca-Cola, Disney, ... các hãng phim lớn nhỏ, các chuyên gia độc lập, sinh viên và những người đam mê.

2.1.3. Tính năng của Engine Unity

Môi trường phát triển được tích hợp với vavv tính năng kế thừa, khả năng chỉnh sửa đồ họa, chức năng kiểm tra chi tiết và đặc biệt tính năng xem trước game ngay trong quá trình xây dựng.

Triển khai được trên nhiều nền tảng:

Chương trình độc lập trên nền tảng Windows và Mac OS

Trên web, thông qua Unity Web Player plugin cho Brower cho cả Window và MacOS

Tài nguyên (Model, âm thanh, hình ảnh, ...) được tải vào trong Unity và tự động cập nhật nếu tài nguyên có sự thay đổi. Unity hỗ trợ các kiểu định dạng từ 3DS Max, Maya, Blender,...

Graphics Engine sử dụng Direct3D (Windows), OpenGL (Mac, Windows),

OpenGL ES (iPhone OS), và các API khác trên Wii.

Unity Asset Server: Đây là một tính năng khá mới của Unity, theo đó Unity sẽ cung cấp một hệ thống quản lý theo dạng phiên bản cho tất cả asset và script. Đây là một kho chứa các tài nguyên cần thiết cho việc làm game. Khi import cũng như sửa chữa, trạng thái của asset ngay lập tức được cập nhật. Server chạy trên database opensource PostgreSQL và có thể truy cập trên cả Mac lẫn Windows, Linux.

2.2. Tổng quan về Game Engine

2.2.1. Khái niệm

Một Game Engine là một phần mềm được viết cho mục đích thiết kế và phát triển video Game. Có rất nhiều loại Game Engine dùng để thiết kế Game cho các hệ máy như hệ Consoles hay máy tính cá nhân (PC). Chức năng cốt lõi của Game Engine phần lớn nằm trong công cụ dựng hình (Kết xuất đồ họa) cho các hình ảnh 2 chiều (2D) hay 3 chiều (3D), công cụ vật lý (hay công cụ tính toán và phát hiện va chạm), âm thanh, mã nguồn, hình ảnh động (Animations), trí tuệ nhân tạo, phân luồng, tạo dòng xử lý dữ liệu, quản lý bộ nhớ, dựng ảnh đồ thị, ... Quá trình phát triển Game tiết kiệm được rất nhiều thời gian và kinh phí vào việc tái sử dụng và tái thích ứng một Engine để tạo ra nhiều Game khác nhau.

2.2.2. Mục đích

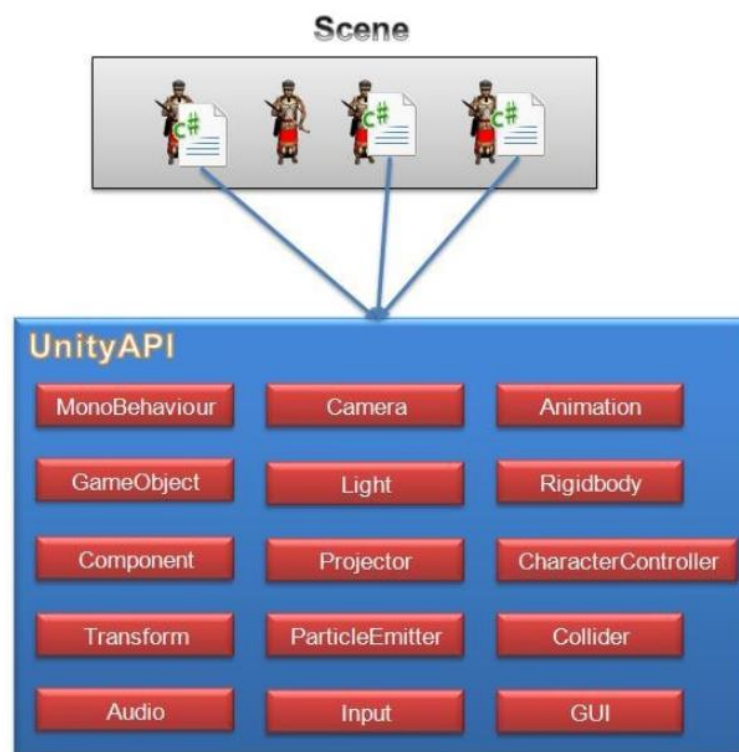
Game Engine cung cấp một bộ các công cụ phát triển trực quan và có thể tái sử dụng từng thành phần trong đó. Nói chung, các bộ công cụ này cung cấp môi trường phát triển tích hợp được đơn giản hóa. Phát triển ứng dụng nhanh (Rapid Application Development) cho Game theo cách lập trình hướng dữ liệu. Những Game Engine này đôi khi còn được gọi là “các phần mềm trung gian cho Game” (Game Middleware), như ý nghĩa của thuật ngữ, chúng cung cấp một nền tảng phần mềm linh hoạt và dễ dàng sử dụng lại với mọi chức năng cốt lõi cần thiết ngay trong nó để có thể phát triển ứng dụng Game đồng thời giảm giá thành, độ phức tạp và kịp thời hạn phát hành – tất cả các yếu tố quan trọng trong ngành Công nghiệp Game đầy tính cạnh tranh. Giống như các phần mềm trung gian khác, Game Engine thường cung cấp một nền tảng trừu tượng hóa, cho phép một Game có thể chạy trên nhiều hệ máy bao gồm các hệ console hoặc máy tính cá nhân với một vài thay đổi (nếu cần) trong mã nguồn của Game đó. Thông thường, phần mềm trung gian cho Game được thiết kế với một nền tảng kiến trúc dựa trên các thành phần khác, cho phép các hệ thống khác nhau trong Engine có thể thay thế hoặc mở rộng các phần mềm trung gian khác chuyên biệt hơn như là Havok cho hệ thống vật lý trong Game, Miles Sound System cho âm thanh hay Bink cho các đoạn video. Một số Game Engine chỉ cung cấp khả năng dựng hình 3D

thời gian thực hay một khả năng riêng biệt nào khác thay vì rất nhiều chức năng trong phạm vi rộng mà Game yêu cầu. Loại Engine này thường được gọi là: “Graphics Engine”, “Rendering Engine”, hay “3D Engine” thay vì thuật ngữ bao quát hơn “Game Engine”

2.3. Tổng quan kiến trúc Game Engine

Engine Unity hỗ trợ trong việc sử dụng UnityAPI để viết script game. UnityAPI là API lập trình game rất mạnh. UnityAPI chứa các đối tượng và phương thức hỗ trợ hầu hết các đối tượng và các loại thành phần trong Unity.

Trong một Scene thường nhiều đối tượng game. Mỗi đối tượng này có thể hoặc có hoặc không có đoạn script nào gắn lên đó. Nếu muốn gắn script vào đối tượng, ta bắt buộc phải kế thừa từ class đó từ lớp **MonoBehaviour** của UnityAPI và tên class phải trùng với tên script. Mỗi script khi gắn lên đối tượng game đều được đối tượng game xem như một thành phần bên trong và được cấp phát vùng nhớ khi chạy game.



Hình 5. Tổng quan kiến trúc UnityAPI

Bên trong UnityAPI chứa nhiều lớp hỗ trợ trong việc lập trình Game, trong đó có một số lớp quan trọng như:

MonoBehaviour: Tất cả các script muốn gắn vào một đối tượng game bắt buộc phải kế thừa từ class này.

GameObject: Lớp cha của tất cả các thực thể trong scene.

Component: Lớp cha của tất cả các thành phần có thể gắn vào đối tượng.

Transform: Giúp thay đổi vị trí, xoay, biến đổi tỷ lệ mô hình.

Input: Hỗ trợ lập trình với chuột, cảm ứng đa điểm, cảm biến gia tốc.

Camera: Giúp lập trình camera.

Light: Giúp tạo ánh sáng trong game.

Project: Giúp chiếu texture lên trên bề mặt vật thể.

ParticleEmitter: Hỗ trợ các hiệu ứng particle đẹp mắt.

Audio: Hỗ trợ lập trình với âm thanh.

Animation: Chạy chuyển động của mô hình nhân vật.

Rigidbody: Giúp tạo hiệu ứng vật lý liên quan đến trọng lực như bóng nảy,...

CharacterController: Giúp điều khiển nhân vật di chuyển theo độ cao địa hình.

Collider: Hỗ trợ lập trình va chạm giữa các vật thể.

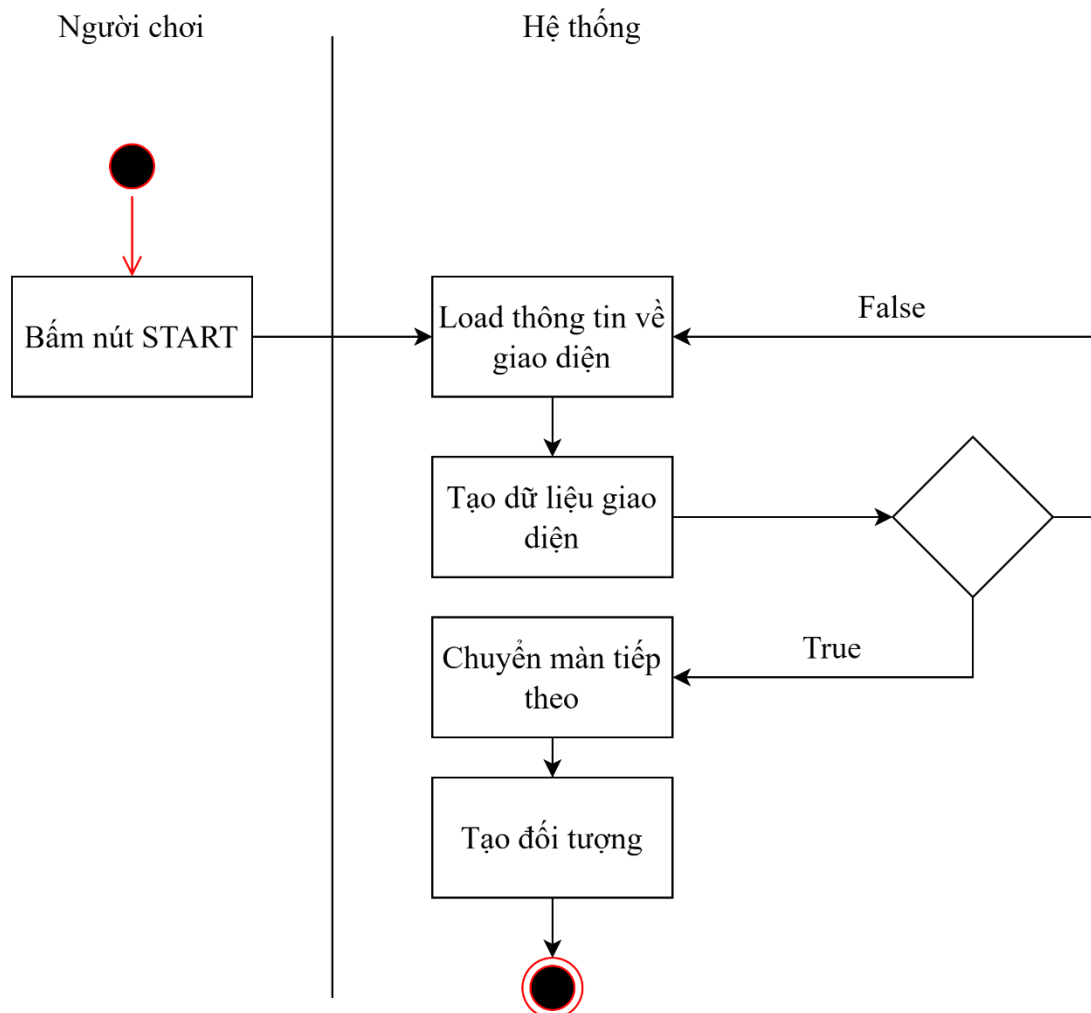
GUI: Giúp lập trình giao diện cho người dùng trên Unity.

CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ

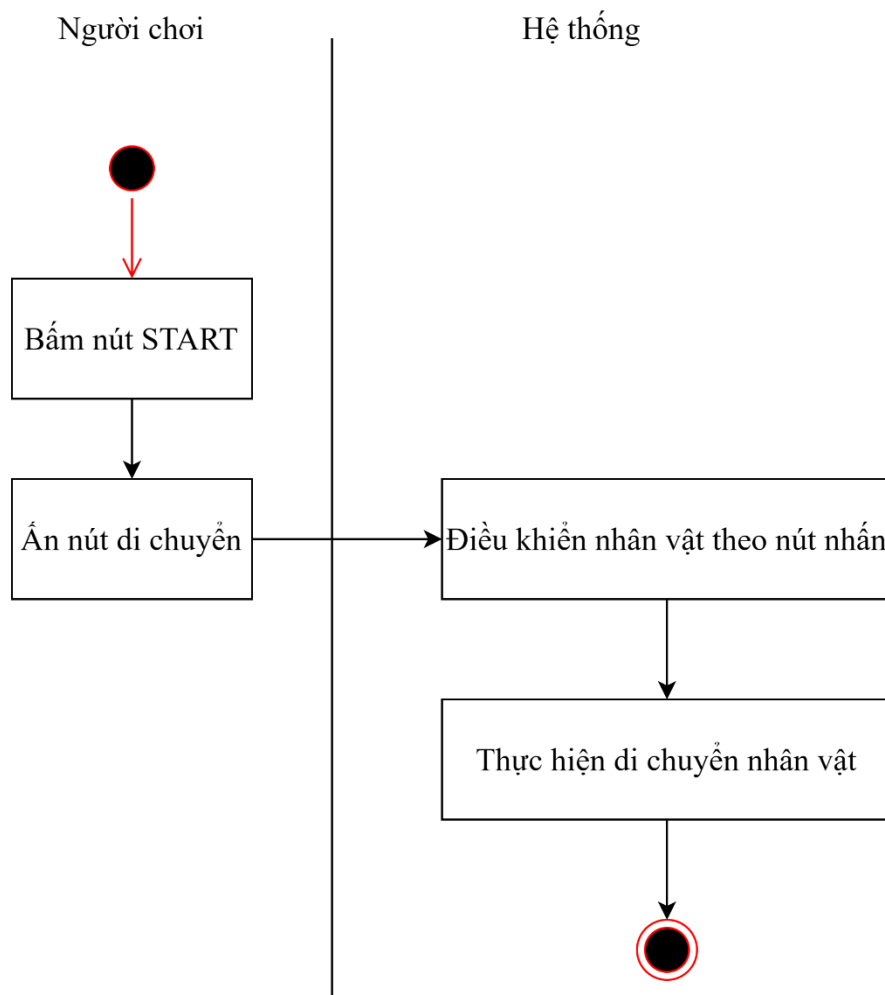
3.1. Các chức năng của hệ thống

Bắt đầu game	Ấn nút START để bắt đầu vào trong game
Điều khiển nhân vật	Sử dụng các nút bấm điều khiển trong môi trường 2D, di chuyển trái, phải, nhảy,...
Phím tắt	Dùng các nút trên bàn phím: <ul style="list-style-type: none">- Nút “space” được dùng để nhảy lên.- Nút “A” (Hoặc mũi tên trái) được dùng để di chuyển nhân vật sang phía bên trái.- Nút “D” (Hoặc mũi tên phải) được dùng để di chuyển nhân vật sang phía bên phải.

3.2. Biểu đồ hoạt động

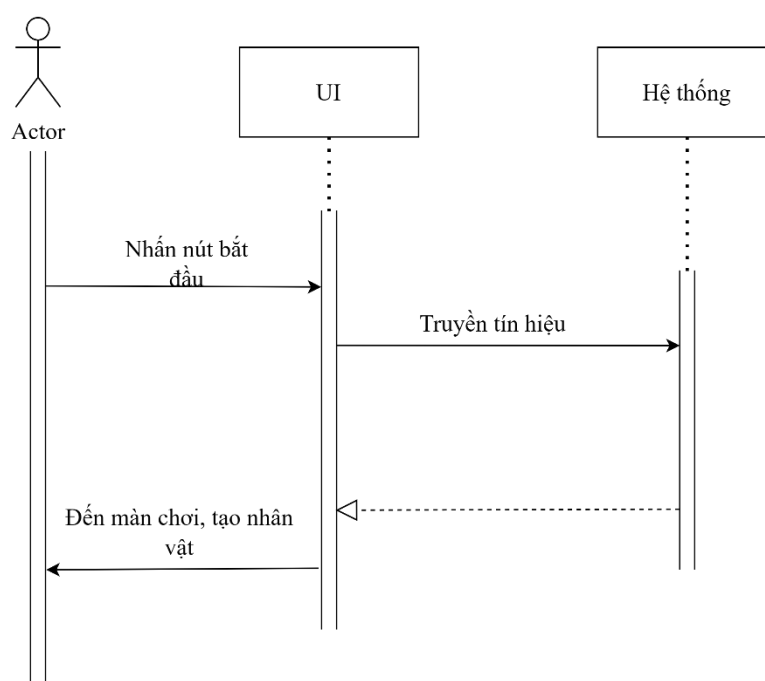


Hình 6. Biểu đồ hoạt động khi bắt đầu game



Hình 7. Biểu đồ điều khiển nhân vật

3.3. Biểu đồ tuần tự



Hình 8. Biểu đồ tuần tự khi bắt đầu chơi game

CHƯƠNG 4. XÂY DỰNG CHƯƠNG TRÌNH

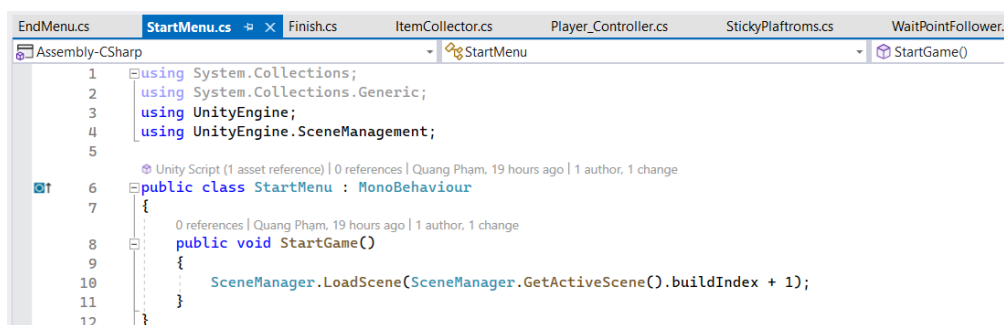
4.1. Xây dựng giao diện

Giao diện đồ họa người dùng là một phần quan trọng không thể thiếu trong khi xây dựng một ứng dụng Game hay bất kỳ một ứng dụng nào để vẽ các đối tượng đồ họa như Button, Label, Image,... lên trên màn hình.

Để có thể làm được như vậy, ta sử dụng lớp GUI (Giao diện đồ họa người dùng). Hệ thống GUI của Unity được gọi là GUIUnity. Để sử dụng được các phương thức trong GUI ta thực hiện gọi từ trong hàm OnGUI() giống như sự kiện trong C#.

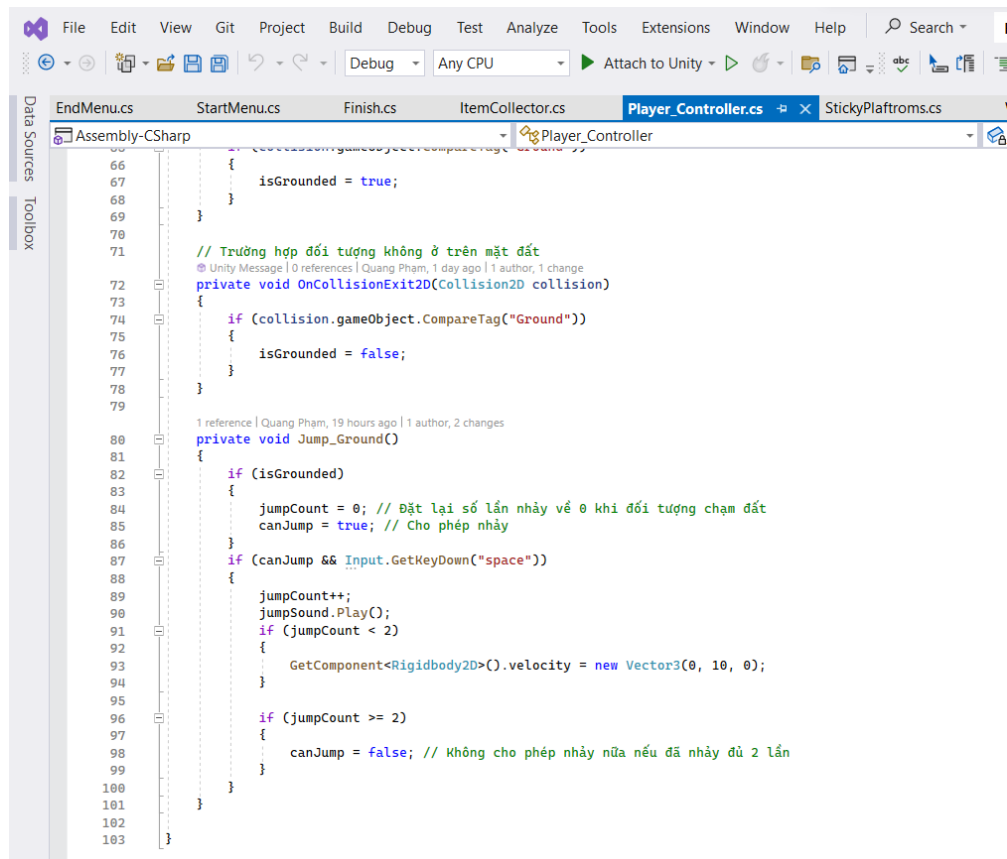


Hình 9. Hình ảnh giao diện ban đầu khi vào Game

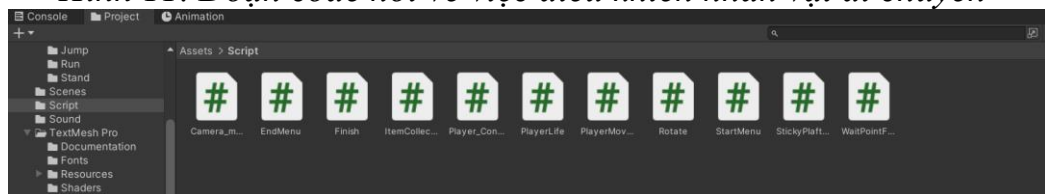


Hình 10. Đoạn code khi người dùng nhấn nút START

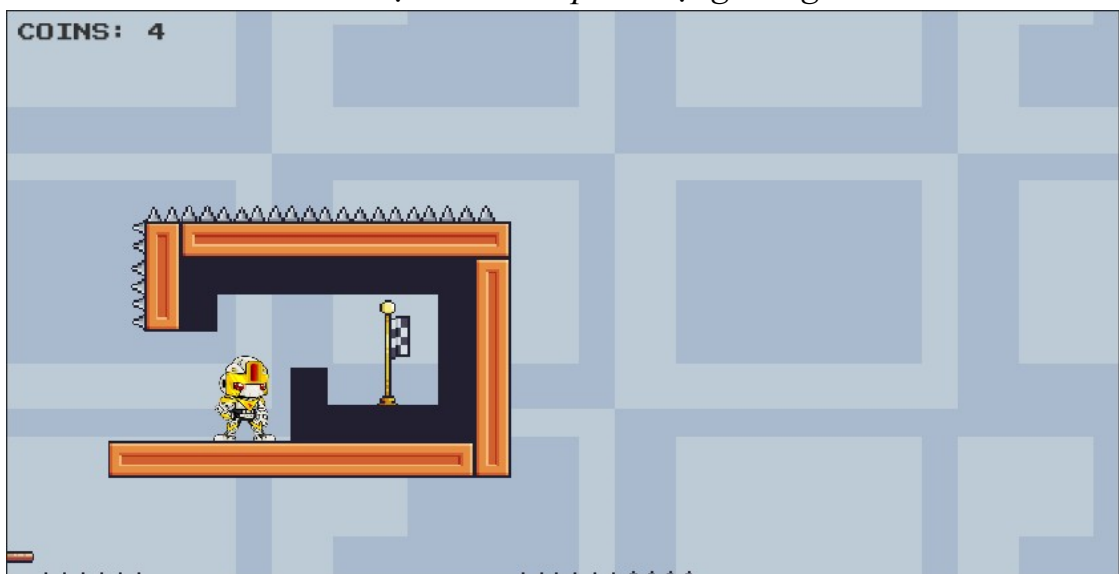
Khi người dùng nhấn nút START, màn hình giao diện sẽ được Load lên. Khi người dùng thực hiện Click vào button (START) sẽ bắt đầu chương trình màn hình khi mở giao diện Game để người dùng có thể bắt đầu chơi Game.



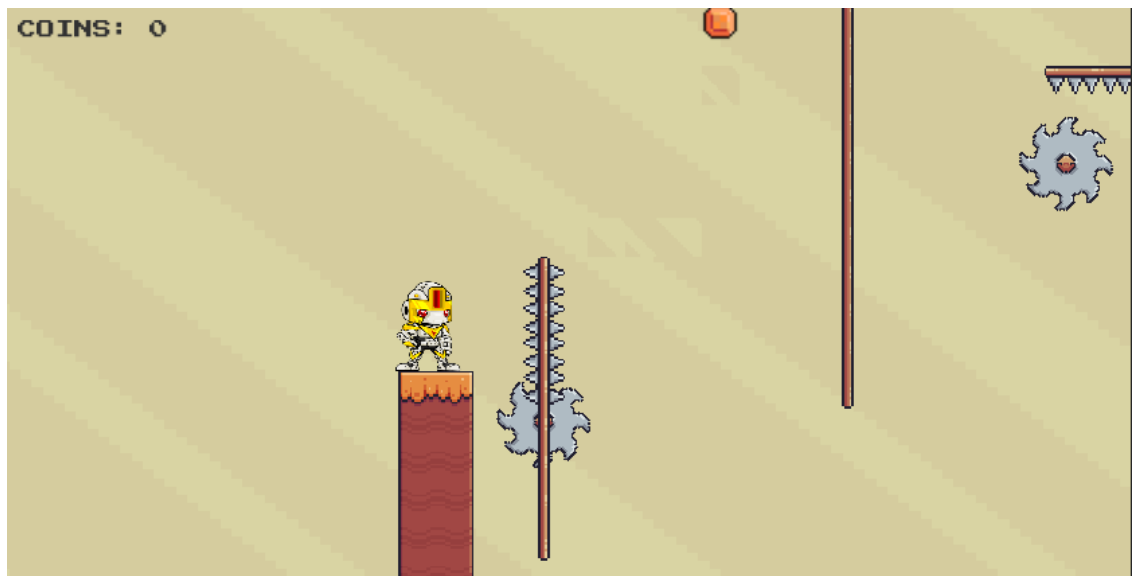
Hình 11. Đoạn code nói về việc điều khiển nhân vật di chuyển



Hình 12. Đoạn code script sử dụng trong Game



Hình 13. Giao diện khi người chơi ở vạch đích màn 1



Hình 14. Giao diện ban đầu khi ở màn 2

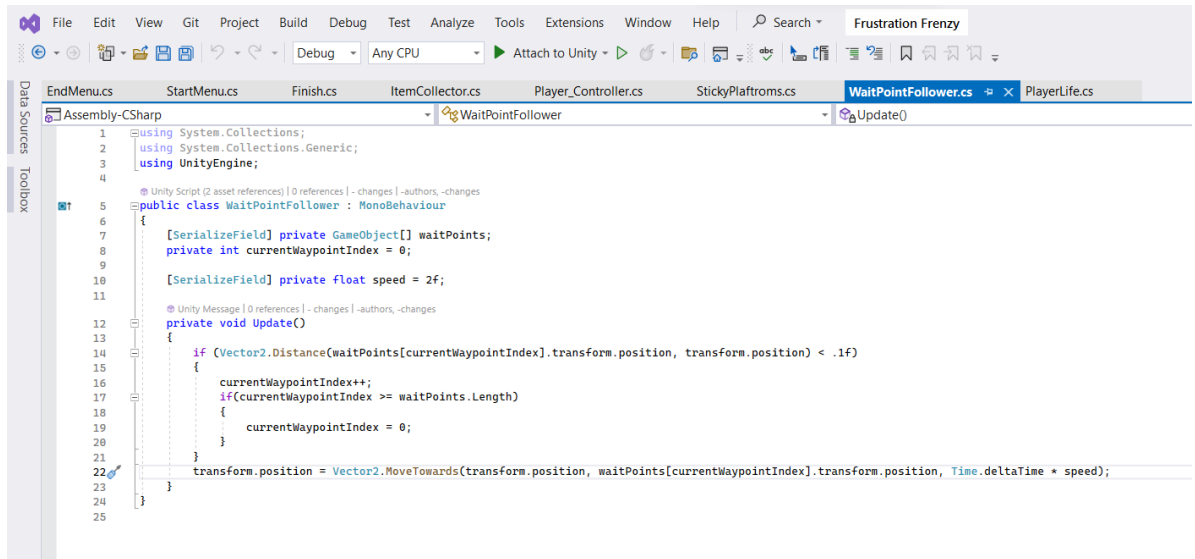


Hình 15. Hình ảnh giao diện màn 2

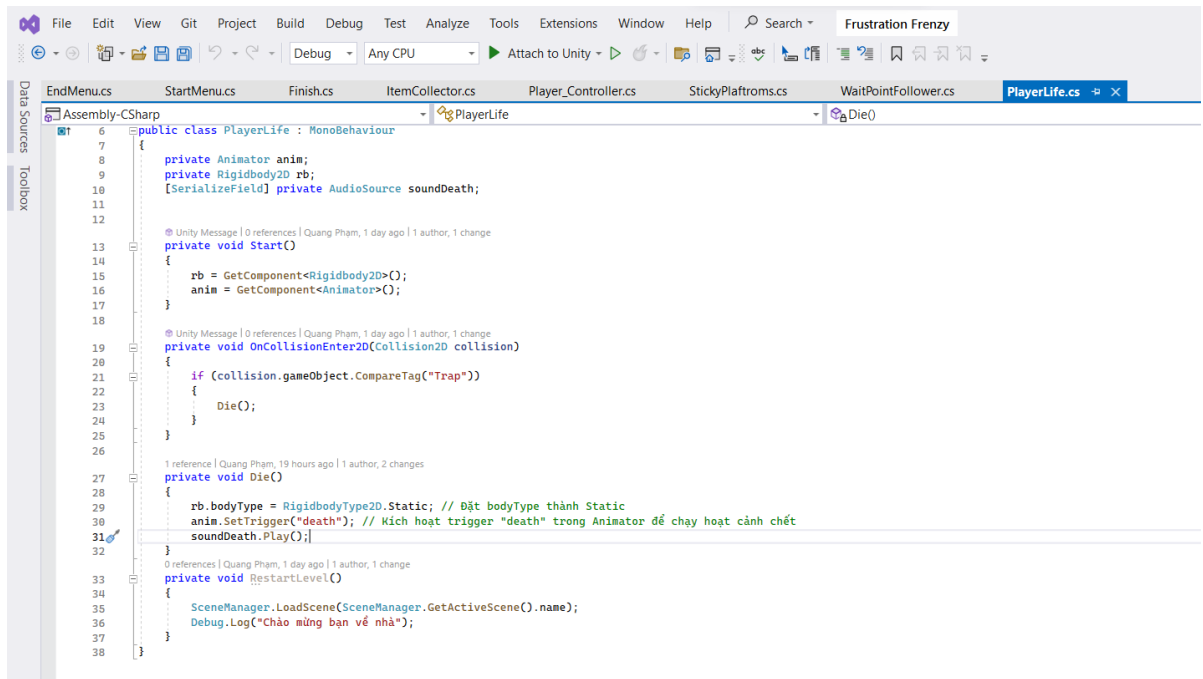


Hình 16. Hình ảnh kết thúc Game

4.2. Code chương trình



Hình 17. Code chương trình lấy điểm di chuyển của vật trong Game



Hình 18. Kiểm tra trạng thái của người chơi

```
4 using UnityEngine.SceneManagement;
5
6 public class Finish : MonoBehaviour
7 {
8     private AudioSource next_LeverSound;
9     private void Start()
10    {
11        next_LeverSound = GetComponent<AudioSource>();
12    }
13
14    private void OnTriggerEnter2D(Collider2D collision)
15    {
16        if(collision.gameObject.name == "Player")
17        {
18            next_LeverSound.Play();
19            Invoke("CompleteLever", 2f);
20        }
21    }
22
23    private void CompleteLever()
24    {
25        // SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
26        int nextSceneIndex = SceneManager.GetActiveScene().buildIndex + 1;
27        int sceneCount = SceneManager.sceneCountInBuildSettings;
28
29        if (nextSceneIndex < sceneCount)
30        {
31            SceneManager.LoadScene(nextSceneIndex);
32        }
33        else
34        {
35            Debug.Log("Đã hoàn thành cấp độ cuối cùng");
36        }
37    }
38 }
39
```

Hình 19. Code chuyển màn hình tiếp theo khi người chơi giành chiến thắng

KẾT LUẬN

Sau thời gian lựa chọn đề tài, cuối cùng em đã chọn đề tài của mình có tên là: “Frustration Frenzy” là đề tài làm game 2D Unity.

Bước đầu tìm hiểu về Unity và Game Engine, em đã gặp rất nhiều khó khăn trong việc tìm kiếm tài liệu và xây dựng ý tưởng Game vì tài liệu về Unity đa phần bằng Tiếng Anh. Nhưng sau một khoảng thời gian cố gắng tìm tòi, học hỏi về việc lập trình game Unity gần hoàn chỉnh để có thể phục vụ cho đề tài này. Trong khoảng thời gian làm đề tài này, em cũng có một số những vấn đề đạt được và chưa đạt được:

Đạt được:

Hiểu được cấu trúc, cách thức hoạt động, cách thức phát triển ứng dụng trên Unity.

Nâng cao kiến thức lập trình và tiếng anh chuyên ngành thông qua những tài liệu về Unity bằng tiếng Anh.

Thực hiện tổ chức Load các màn tiếp theo tương ứng khi người chơi qua màn trước đó.

Xây dựng chức năng qua màn, tính điểm thông qua việc thu thập coin, ... thông qua ngôn ngữ lập trình C# trên Visual Studio 2022.

Tạo âm thanh cho trò chơi và âm thanh tùy theo từng trường hợp người dùng điều khiển nhân vật.

Chưa đạt được:

Do kiến thức của em về Unity còn hạn hẹp nên các chức năng trong game vẫn còn nghèo nàn.

Chưa xây dựng được hoàn chỉnh 1 game như mong muốn.

Các Script trong Unity tổ chức chưa thực sự rõ ràng.

Sắp xếp nhân vật và màn chơi chưa được Logic.

Thuận lợi:

Vận dụng các kiến thức đã học trong trường để làm đề tài này.

Được sự chỉ dạy tận tình của thầy và các bạn.

Tài liệu có nhiều trên Internet

Khó khăn:

Unity là một môi trường lập trình tương đối mới đối với em nên khi bắt tay vào làm thì còn nhiều bỡ ngỡ.

TÀI LIỆU THAM KHẢO

[1]. Ryan Henson Creighton, *Unity 3D Game Development by Example*, Packt publishing, 2010.

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

[3]. Link GitHub:

https://github.com/JohnsonAudreyRalph/GameUnity_End_of_Course.git