

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**---------------------------------------**

**BÁO CÁO THÍ NGHIỆM/THỤC NGHIỆM**

**PHÁT TRIỂN ỨNG DỤNG GAME**

XÂY DỰNG ỨNG DỤNG GAME 2D PLATFORMER

“Freight Transportation”

| **GVHD:** | **TS.Vũ Đức Huy** |
| --- | --- |
| **Sinh viên:** |  |
|  | **Đỗ Tiến Tùng-2021606459** |
|  | **Trần Trọng Hoàn** |
|  | **Hoàng Trung Nam** |
|  | **Đinh Đức Anh** |
|  | **Trần Xuân Quang** |
| **Nhóm:** | **16** |
| **Lớp:** | **20241IT6028003. Khoá: K16** |

**Hà Nội – Năm 2024**

**Mục lục**

[Lời cảm ơn 9](#_heading=h.30j0zll)

[Lời mở đầu: 10](#_heading=h.1fob9te)

[1. Lý do chọn đề tài 10](#_heading=h.3znysh7)

[2. Mục tiêu của đề tài 10](#_heading=h.2et92p0)

[3. Đối tượng nghiên cứu 10](#_heading=h.tyjcwt)

[4. Phạm vi nghiên cứu 10](#_heading=h.3dy6vkm)

[CHƯƠNG 1. TỔNG QUAN VÀ CƠ SỞ LÝ THUYẾT 12](#_heading=h.1t3h5sf)

[1.1](#_heading=h.4d34og8) TỔNG QUAN VỀ ENGINE UNITY 12

[1.1.1](#_heading=h.2s8eyo1) Giới thiệu 12

[1.1.2](#_heading=h.17dp8vu) Sơ lược lịch sử hình thành và phát triển của Unity 13

[1.1.3](#_heading=h.3rdcrjn) Tính năng của ENGINE UNITY 13

[1.2. TỔNG QUAN VỀ CÁC THÀNH PHẦN TRONG UNITY 14](#_heading=h.26in1rg)

[1.2.1](#_heading=h.lnxbz9) Assets 14

[1.2.2](#_heading=h.35nkun2) Scenes 15

[1.2.3](#_heading=h.1ksv4uv) Game Object 15

[1.2.4](#_heading=h.44sinio) Components 15

[1.2.5](#_heading=h.2jxsxqh) Scripts 15

[1.2.6](#_heading=h.z337ya) Prefabs 16

[1.2.7](#_heading=h.3j2qqm3) Collider 16

[1.2.8](#_heading=h.1y810tw) Rigidbody 17

[1.2.9](#_heading=h.4i7ojhp) Sprite 18

[1.2.10 Animator 18](#_heading=h.2xcytpi)

[1.2.11 Audio Source 18](#_heading=h.1ci93xb)

[1.2.12 Camera 19](#_heading=h.3whwml4)

[1.2.13 Transform 20](#_heading=h.2bn6wsx)

[1.2.14 Renderer 20](#_heading=h.qsh70q)

[1.3](#_heading=h.3as4poj) TỔNG QUAN VỀ VISUAL STUDIO 20

[1.3.1](#_heading=h.1pxezwc) Giới thiệu 20

[1.3.2](#_heading=h.49x2ik5) Tính năng 21

[CHƯƠNG 2.PHÁT TRIỂN KỊCH BẢN VÀ THIẾT KẾ GAME 24](#_heading=h.2p2csry)

[2.1. Giới thiệu tổng quan 24](#_heading=h.147n2zr)

[2.1.1. Thông tin game 24](#_heading=h.3o7alnk)

[2.1.2. Thể loại game và yếu tố game 24](#_heading=h.23ckvvd)

[2.1.3. Đối tượng chơi 25](#_heading=h.ihv636)

[2.1.4. Nền tảng 25](#_heading=h.32hioqz)

[2.2. Kịch bản game 25](#_heading=h.1hmsyys)

[2.2.1. Mô tả 25](#_heading=h.41mghml)

[2.2.2. Luật chơi 26](#_heading=h.2grqrue)

[2.2.3. Tính điểm 26](#_heading=h.vx1227)

[2.2.4. Tương tác và điều khiển game 27](#_heading=h.3fwokq0)

[2.2.5. Các phần tử của game (Game Element) 27](#_heading=h.1v1yuxt)

[2.2.5.1. Player 27](#_heading=h.4f1mdlm)

[2.2.5.2 Player skill: 28](#_heading=h.19c6y18)

[2.2.5.3. Enemy và boss 30](#_heading=h.2zbgiuw)

[2.2.6. Thiết kế các lv của game 32](#_heading=h.1rvwp1q)

[2.2.7. Game Mechanic 33](#_heading=h.4bvk7pj)

[2.3 Thiết kế giao diện 33](#_heading=h.2r0uhxc)

[2.3.1. Menu khởi đầu 33](#_heading=h.1664s55)

[2.3.2. Menu chọn map 34](#_heading=h.25b2l0r)

[2.3.3. Màn hình Tutorial 34](#_heading=h.34g0dwd)

[2.3.4: UI trong game 35](#_heading=h.43ky6rz)

[2.3.5. Màn hình Pause 35](#_heading=h.xvir7l)

[2.3.6. Màn hình chiến thắng: 36](#_heading=h.1x0gk37)

[2.3.7. Màn hình thua trận 36](#_heading=h.1baon6m)

[2.3.8 : Story Board 37](#_heading=h.pkwqa1)

[2.4. Tài nguyên: 37](#_heading=h.1opuj5n)

[2.4.1. Hình ảnh: 37](#_heading=h.48pi1tg)

[2.4.2. Âm thanh 38](#_heading=h.1302m92)

[2.4.3. Game Assest 38](#_heading=h.3mzq4wv)

[2.4.4. Tham khảo tutorial: 38](#_heading=h.2250f4o)

[CHƯƠNG 3: CÁC KĨ THUẬT XÂY DỰNG GAME VÀ KẾT QUẢ THỰC NGHIỆM. 39](#_heading=h.haapch)

[3.1. Các kĩ thuật xây dựng game 39](#_heading=h.319y80a)

[3.1.1 Các kĩ thuật với Player 39](#_heading=h.1gf8i83)

[3.1.1.1 Di chuyển và chạy 39](#_heading=h.40ew0vw)

[3.1.1.2. Camera theo dõi player 40](#_heading=h.3ep43zb)

[3.1.1.3. Thể trạng nhân vật (Áp dụng cho cả enemy/boss) 40](#_heading=h.4du1wux)

[3.1.1.4. Tấn công 41](#_heading=h.184mhaj)

[3.1.1.5. Xử lý va chạm với colider khác 43](#_heading=h.meukdy)

[3.1.2. Các kĩ thuật đối với enemy/boss. 43](#_heading=h.1ljsd9k)

[3.1.2.1. Các hành vì của các đối tượng 43](#_heading=h.45jfvxd)

[3.1.2.2. Xử lý va chạm tấn công tầm gần: 46](#_heading=h.1yyy98l)

[3.1.2.3. Xử lý projectiles bay 46](#_heading=h.2y3w247)

[3.1.2.4. Xử lý va chạm projectiles: 46](#_heading=h.3x8tuzt)

[3.1.2.5. Kỹ thuật xây dựng spawner 47](#_heading=h.rjefff)

[3.1.3. Xây dựng timer. 48](#_heading=h.2pta16n)

[3.1.4. Kỹ thuật đếm các kẻ địch đã bị tiêu diệt, số hit nhân vật đã nhận và xử lý điều kiện chiến thắng 48](#_heading=h.3oy7u29)

[3.1.5. Kỹ thuật tính điểm. 50](#_heading=h.1idq7dh)

[3.1.6 Kỹ thuật thiết kế map 51](#_heading=h.2hio093)

[3.2 Kết quả thực nghiệm 51](#_heading=h.3gnlt4p)

[3.2.1. Main menu 52](#_heading=h.1vsw3ci)

[3.2.2. Màn hình game 53](#_heading=h.3u2rp3q)

[3.2.3. Màn hình chiến thắng và thua 54](#_heading=h.38czs75)

[KẾT LUẬN 56](#_heading=h.2mn7vak)

[TÀI LIỆU THAM KHẢO 58](#_heading=h.11si5id)

**Danh mục hình ảnh**

[Hình 2.1: Nhân vật chính 27](#_heading=h.2u6wntf)

[Hình 2.2: Combo A1 28](#_heading=h.3tbugp1)

[Hình 2.3: Combo A2 28](#_heading=h.28h4qwu)

[Hình 2.4: Combo A3 28](#_heading=h.nmf14n)

[Hình 2.5: Combo B1 29](#_heading=h.37m2jsg)

[Hình 2.6: Combo B2 29](#_heading=h.1mrcu09)

[Hình 2.7: Combo B3 hit 1 29](#_heading=h.46r0co2)

[Hình 2.8: Combo B3 hit 2 29](#_heading=h.2lwamvv)

[Hình 2.9: Combo B3 hit 3 29](#_heading=h.3l18frh)

[Hình 2.10: Combo B4 30](#_heading=h.206ipza)

[Hình 2.11: Aerial attack 30](#_heading=h.4k668n3)

[Hình 2.12: Demon Bee 31](#_heading=h.1egqt2p)

[Hình 2.12: Goblin 31](#_heading=h.3ygebqi)

[Hình 2.13: FireWorm 31](#_heading=h.2dlolyb)

[Hình 2.14: Flying Demon 31](#_heading=h.sqyw64)

[Hình 2.15: Cruel Demon Slime 32](#_heading=h.3cqmetx)

[Hình 2.16: Hình dung màn hình menu chính 33](#_heading=h.3q5sasy)

[Hình 2.17: Hình dung màn hình Menu chọn map 34](#_heading=h.kgcv8k)

[Hình 2.18: Hình dung màn hình Tutorial 34](#_heading=h.1jlao46)

[Hình 2.19 Hình dung màn hình UI 35](#_heading=h.2iq8gzs)

[Hình 2.20: Hình dung màn hình Pause Menu 35](#_heading=h.3hv69ve)

[Hình 2.21: Hình dung màn hình chiến thắng 36](#_heading=h.2w5ecyt)

[Hình 2.22: Hình dung màn hình thua cuộc 36](#_heading=h.2afmg28)

[Hình 2.23: Story Board 37](#_heading=h.39kk8xu)

[Hình 2.24: Ví dụ hình ảnh 38](#_heading=h.2nusc19)

[Hình 3.1: Code thay đổi trạng thái 39](#_heading=h.2fk6b3p)

[Hình 3.2: Code phím thay đổi trạng thái 39](#_heading=h.upglbi)

[Hình 3.3: Cinemachine 40](#_heading=h.1tuee74)

[Hình 3.4: Máu và trạng thái sống của nhân vật 41](#_heading=h.2szc72q)

[Hình 3.5: Input tác động vào animation 42](#_heading=h.3s49zyc)

[Hình 3.6: ComboA flowchart 42](#_heading=h.279ka65)

[Hình 3.7: Xử lý va chạm collider của Hitbox 43](#_heading=h.36ei31r)

[Hình 3.8: Player Tracking Scripts 44](#_heading=h.2koq656)

[Hình 3.8:Nhận diện collider của “Player” trong tầm detect. 45](#_heading=h.zu0gcz)

[Hình 3.9: Nếu có “Player” trong tầm thì sẽ ra đòn 45](#_heading=h.3jtnz0s)

[Hình 3.10: Xử lý va chạm collider của Hitbox 46](#_heading=h.4iylrwe)

[Hình 3.11: Projectiles bay 46](#_heading=h.1d96cc0)

[Hình 3.11: Xử lý va chạm projectiles với collider 46](#_heading=h.2ce457m)

[Hình 3.12: Tạo list chứa các prefabs quái 47](#_heading=h.3bj1y38)

[Hình 3.7: Generate Enemy 47](#_heading=h.1qoc8b1)

[Hình 3.14: Thiết lập thời gian giữa từng wave 47](#_heading=h.4anzqyu)

[Hình 3.15: Timer 48](#_heading=h.14ykbeg)

[Hình 3.16: Khởi tạo bộ đếm + condition 49](#_heading=h.243i4a2)

[Hình 3.17: Địch chết tăng bộ đếm địch 49](#_heading=h.j8sehv)

[Hình 3.18: Nhân vật nhận đòn tự động tăng bộ đếm 50](#_heading=h.338fx5o)

[Hình 3.19: Tính điểm 50](#_heading=h.42ddq1a)

[Hình 3.20: Tile map 51](#_heading=h.wnyagw)

[Hình 3.21: Main menu 52](#_heading=h.4fsjm0b)

[Hình 3.21: Tutorial 52](#_heading=h.2uxtw84)

[Hình 3.22: Select stage 53](#_heading=h.1a346fx)

[Hình 3.23: Màn hình Ingame 53](#_heading=h.2981zbj)

[Hình 3.24: Màn hình pause 54](#_heading=h.odc9jc)

[Hình 3.25: Màn hình chiến thắng và tính điểm 54](#_heading=h.1nia2ey)

[Hình 3.26: Màn hình thua 55](#_heading=h.47hxl2r)

# Lời cảm ơn

Em xin trân trọng cảm ơn thầy Vũ Đức Huy đã tận tình chỉ dạy, cầm tay chỉ việc và tạo mọi điều kiện thuận lợi cho chúng em trong suốt quá trình học tập và hoàn thành đồ án tốt nghiệp.

Trong quá trình hoàn thành bài tập lớn không thể tránh khỏi những thiếu sót. Kính mong thầy cô chỉ bảo và đóng góp ý kiến để đề tài đồ án của em được hoàn thiện hơn.

Lời cuối cùng em xin chúc các quý thầy, cô và các bạn luôn dồi dào sức khỏe và thành công trong cuộc sống. Xin chúc những điều tốt đẹp nhất sẽ luôn đồng hành cùng mọi người!

Em xin chân thành cảm ơn.

# Lời mở đầu:

## 1. Lý do chọn đề tài

Ngành công nghiệp game giải trí đang ngày càng phát triển mạnh mẽ. Nhằm đáp ứng nhu cầu giải trí của giới trẻ trong thời đại số hiện nay. Đây cũng là lĩnh vực yêu cầu sự đầu tư nghiên cứu và tư duy lập trình như phát triển phần mềm, website. Vì vậy bản thân em đã quyết định triển khai đề tài game của mình: “*Freight Transportation*”. Đề tài này giúp em thử sức và trang bị cho bản thân những kiến thức và quy trình về lập trình game.

## 2. Mục tiêu của đề tài

Tận dụng các kiến thức đã trang bị trong toàn bộ khoá học ở trường và giải quyết đề tài làm game này.

Có khả năng giải quyết đề tài làm game này bằng máy tính và viết báo cáo về vấn đề thực hiện.

Khảo sát, phân tích, tự học, tự nghiên cứu cách xây dựng, thiết kế game và lập trình game 2D trong Unity, sau đó kiểm thử sản phẩm; tác phong làm việc chuyên nghiệp và tuân thủ đạo đức nghề nghiệp.

## 3. Đối tượng nghiên cứu

Sử dụng ngôn ngữ lập trình C# và kiến thức về phần mềm Unity để xây dựng, thiết kế game.

Trong đề tài kết hợp sử dụng các phần mềm chính sau để thực hiện:

* Phần mềm Unity version 2022.3.21f1
* Phần mềm Visual Studio 2019

## 4. Phạm vi nghiên cứu

Phạm vi nội dung: tập trung nghiên cứu cách xây dựng, thiết kế game và lập trình game 2D trong Unity, rồi tiến hành kiểm thử game.

Phạm vi thời gian: đề tài được thực hiện trong khoảng 10 tuần từ ngày 15/09/2024 đến 20/12/2024.

Ngoài các phần Mở đầu, Kết luận và Tài liệu tham khảo, báo cáo đồ án gồm 3 chương chính sau.

* Chương 1: Tổng quan và cơ sở lý thuyết
* Chương 2: Phát triển kịch bản và thiết kế game
* Chương 3: Các kỹ thuật xây dựng game và kết quả thực nghiệm.

# CHƯƠNG 1. TỔNG QUAN VÀ CƠ SỞ LÝ THUYẾT

## 1.1 TỔNG QUAN VỀ ENGINE UNITY

### 1.1.1 Giới thiệu

Unity mang lại sức mạnh kỳ diệu cho nhân vật mà chúng ta muốn thể hiện sống động hơn trong không gian 3 chiều huyền ảo. Công nghệ cao này tạo ra 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.

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 hao hao với Director, Blender game engine, Virtools hay Torque Game Builder trong khía cạnh dùng môi trường đồ họa tích 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. Trình biên tập có thể chạy trên Windows và Mac OS, và có thể xuất ra game cho Windows, Mac, Wii, iOS, Android. Game cũng có thể chơi trên trình duyệt web thông qua plugin Unity Web Player. Unity mới bổ sung khả năng xuất ra game trên widget cho Mac, và cả Xbox 360, PlayStation 3.

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 hầu hết trên 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 là thật tuyệt, chúng ta sẽ có một lợi thế lớn khi viết game trên Unity này, bởi công 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 sẽ nhanh 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 import 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. Ngoài ra Unity còn cho chúng ta trực tiếp tạo các mô hình nếu muốn. Việc đặt các thuộc tính vật lý trong Unity cũng cực kỳ dễ dàng và hỗ trợ sẵn nhiều chức năng.

### 1.1.2 Sơ lược lịch sử hình thành và phát triển của Unity

Phần lõi của Unity ban đầu được viết bởi Joachim Ante vào năm 2001. Sau đó công ty được 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 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 Unity 5. Có thể thấy tốc độ phát triển của Unity khá nhanh.

Unity được trên 250.000 người đăng ký sử dụng gồm Bigpoint, Cartoon Network, Coca-Cola, Disney, Electronic Arts, LEGO, Microsoft, NASA, Ubisoft, Warner Bros, các hãng phim lớn nhỏ, các chuyên gia độc lập, sinh viên và những người đam mê.

### 1.1.3 Tính năng của ENGINE UNITY

Môi trường phát triển được tích hợp với 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 lúc xây dựng (live game preview).

Triển khai được trên nhiều nền tảng:

Chương trình độc lập trên Windows và Mac OS.

Trên web, thông qua Unity Web Player plugin cho Internet Explorer, Firefox, Safari, Opera, Chrome, cho cả Windows và Mac OS.

Trên Mac OS Dashboard widget.

Cho Nintendo Wii (cần mua license thêm.)

Cho iPhone, iPad application (cần mua license thêm.) Cho Google Android (cần mua license thêm.)

Cho Microsoft Xbox 360 (cần mua license thêm.) Cho Sony PlayStation 3 (cần mua license thêm.)

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, Cinema 4D và Cheetah3D.

Graphics engine sử dụng Direct3D (Windows), OpenGL (Mac, Windows), OpenGL ES (iPhone OS), và các API khác trên Wii.

Hỗ trợ bump mapping, reflection mapping, parallax mapping, Screen Space Ambient Occlusion v...v...

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 cả 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 open source PostgreSQL và có thể truy cập trên cả Mac lẫn Windows, Linux. Asset Server.

## 1.2. TỔNG QUAN VỀ CÁC THÀNH PHẦN TRONG UNITY

### 1.2.1 Assets

Assets là tài nguyên xây dựng nên một dự án trên Unity. Những tài nguyên có thể là hình ảnh, âm thanh, mô hình 2D 3D, chất liệu (material), texture… hoặc cả một project hoàn chỉnh.

Các asset do chính những nhà phát triển game tạo ra và có thể được download miễn phí hoặc trả phí trên Unity Asset Store. Đây là một trong những tính năng rất hay của Unity. Các asset này sẽ giúp giảm thiểu rất nhiều thời gian cho việc thiết kế và lập trình game.

### 1.2.2 Scenes

Trong Unity, một cảnh chơi (hoặc một phân đoạn) là những màn chơi riêng biệt, một khu vực trong game hoặc thành phần có trong nội dung của trò chơi (các menu). Các thành phần này được gọi là Scene. Bằng cách tạo ra nhiều Scenes, chúng ta có thể phân phối thời gian và tối ưu tài nguyên, kiểm tra các phân đoạn trong game một cách độc lập.

### 1.2.3 Game Object

Khi Asset được sử dụng trong các Scene, Unity định nghĩa đó là Game Object. Đây là một thuật ngữ thông dụng, đặc biệt trong mảng lập trình. Tất cả các Game Object đều chứa ít nhất một thành phần cơ bản là Transform, lưu trữ thông tin về vị trí, góc xoay và tỉ lệ của Game Object. Thành phần Transform có thể được tuỳ biến và chỉnh sửa trong quá trình lập trình.

### 1.2.4 Components

Components là các thành phần trong game, bổ sung tính năng cho các Game Object. Mỗi Component có chức năng riêng biệt. Đa phần các Component phụ thuộc vào Transform, vì nó lưu trữ các thông số cơ bản của Game Object. Bản chất của Game Object là không có gì cả, các đặc tính và khả năng của Gameobject nằm hoàn toàn trong các Component. Do đó chúng ta có thể xây dựng nên bất kỳ Game Object nào trong game mà chúng ta có thể tưởng tượng được.

### 1.2.5 Scripts

Scripts được Unity xem như một Component. Đây là thành phần thiết yếu trong quá trình phát triển game. Bất kỳ một game nào, dù đơn giản nhất đều cần đến Scripts để tương tác với các thao tác 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 các ngôn ngữ: JavaScript, C#. Unity không đòi hỏi lập trình viên phải học cách lập trình trong Unity, nhưng trong nhiều tình huống, chúng ta cần sử dụng Script trong mỗi phần của kịch bản game.

Để viết Script, chúng ta có thể làm việc với một trình biên tập Script độc lập của Unity, hoặc làm việc trên Mono Developer được tích hợp vào Unity trong những phiên bản gần đây. Mono Developer là một IDE khá tốt, cung cấp nhiều chức năng tương tự Visual Studio chúng ta cũng có thể dùng Visual Studio để viết file C# như bình thường . Mã nguồn viết trên Mono Developer sẽ được cập nhật và lưu trữ trong dự án trên Unity.

### 1.2.6 Prefabs

Prefabs thực chất là Game Object được lưu trữ lại để tái sử dụng. Các Game Object được nhân bản từ một prefab sẽ giống nhau hoàn toàn, ngoại trừ thành phần Transform để phân biệt và quản lý được tốt hơn.

Để tạo ra một prefab, ta đơn giản chỉ cần kéo một Game Object vào cửa sổ Project.

### 1.2.7 Collider

Hệ thống xử lý va chạm bào gồm 2D và 3D. Được chia ra các va chạm của các hình cơ bản:

Box collider: Va chạm cho các vật hình hộp chữ nhật đối với 3D và hình chữ nhật đối với 2D.

* Is Trigger: dạng true/false, cho phép va chạm có đi xuyên qua không?
* Material: tham chiếu đến physic material.
* Center: điều chỉnh thông số tâm của va chậm theo các trục tương ứng.
* Size: điều chỉnh kích thước theo các trục tương ứng.

Capsule Collider: Va chạm dành cho các khối hình trụ.

* Is Trigger: dạng true/false, cho phép va chạm có đi xuyên qua không?
* Material: tham chiếu đến physic material.
* Center: điều chỉnh thông số tâm của va chạm theo các trục tương ứng.
* Radius: điều chỉnh kích thước bán kính.
* Height: độ cao
* Direction: xoay hình trụ theo các trục tương ứng.Box collider: Va chạm cho các vật hình hộp chữ nhật đối với 3D và hình chữ nhật đối với 2D.

Mesh Collider: Va chạm dạng lưới dành các vật thể không xác định hình thể.

Sphere Collider: va chạm áp dụng cho các vật hình cầu.

Wheel collider: dành cho vật thể hình bánh xe. Nó sẽ mô phỏng hệ thống va chạm giống với các bánh xe.

Terrain Collider: dành cho các vật thể địa hình và terrain collider sẽ dựa theo địa hình đó.

### 1.2.8 Rigidbody

Hệ thống mô phỏng vật lý trong game. Cũng chia ra 2D

* Mass: khối lượng (đơn vị tùy ý).
* Drag: Sức cản không khí ảnh hưởng như thế nào với đối tượng khi di chuyển. 0 có nghĩa là không có sức cản không khí, và vô cùng làm cho các đối tượng di chuyển ngay lập tức dừng lại.
* Angular Drag: Sức cản không khí ảnh hưởng đến các đối tượng khi quay từ mô-men xoắn. 0 có nghĩa là không có sức cản không khí. Không thể làm cho vật dừng quay hẳn chỉ bằng cách thiết lập Angular Drag của nó đến vô cùng.
* Use Gravity: Nếu được kích hoạt, các đối tượng bị ảnh hưởng bởi lực hấp dẫn.
* Is Kinematic: Nếu được kích hoạt, các đối tượng sẽ không được thúc đẩy bởi động cơ vật lý.
* Interpolate: giảm xóc.
* Collision detection: Được sử dụng để ngăn chặn các đối tượng chuyển động nhanh qua các đối tượng khác mà không phát hiện va chạm.
* Constraints: Ràng buộc Những hạn chế về chuyển động của Rigidbody.

### 1.2.9 Sprite

Là một hình ảnh 2D của một gameobject có thể là hình ảnh đầy đủ, hoặc có thể là một bộ phận nào đó. Unity cho phép tùy chỉnh màu sắc, kích thước, độ phân giải của một hình ảnh 2D.

### 1.2.10 Animator

Trong 2D thì animation là tập một hình ảnh động dựa trên sự thay đổi liên tục của nhiều sprite khác nhau. Trong 3D là một tập hợp các sự thay đổi theo thời gian của đối tượng trong không gian. Mỗi thay đổi là một keyframe. Key Frame hay Iframe là một trạng thái của một animation. Animator gồm các thành phần:

Controller: Bộ điều khiển animation gắn liền với nhân vật này. Nó sẽ quản lý các animation clip, các thông số tốc độ của animation, thứ tự các clip…

Avatar: thành phần tạo hình ảnh cho object.

Apply Root Motion: dạng true-false, cho phép thiết lập animation có di chuyển theo không gian đã được tạo khi cấu hình animation.

Animate Physics: dạng true-false, khi được chọn, các hình ảnh trong animation sẽ có thể tương tác vật lý với nhau.

Culling mode: chọn chế độ cho hình ảnh động.

### 1.2.11 Audio Source

Âm thanh trong game. Gồm cả âm thanh 2D và 3D.

* Audio clip: tham chiếu đến file âm thanh.
* Mute: chơi ở chế độ như tắt tiếng.
* Bypass effects: bộ lọc hiệu ứng áp dụng cho các nguồn âm thanh.
* Bypass listener effects: Điều này là để nhanh chóng chuyển tất cả các hiệu ứng Listener on / off.
* Pass Reverb Zones: Điều này là để nhanh chóng chuyển tất cả các khu Reverb on / off.
* Play on awake: Nếu được kích hoạt, âm thanh sẽ bắt đầu chơi lúc cảnh ra mắt. Nếu vô hiệu hóa, cần phải bắt đầu nó bằng cách sử dụng lệnh Play () từ kịch bản script.
* Loop: Kích hoạt tính năng này để làm cho Clip âm thanh lặp lại.
* Pitch: Xác định ưu tiên của nguồn âm thanh này trong số tất cả những nguồn âm cùng tồn tại trong bối cảnh đó. (Ưu tiên: 0 = quan trọng nhất, 256 = ít quan trọng nhất Mặc định = 128).
* 3D Sound Setting: Cài đặt được áp dụng cho các nguồn âm thanh nếu Audio Clip là một âm thanh 3D.
* 2D Sound Setting: Cài đặt được áp dụng cho các nguồn âm thanh nếu Audio Clip là một âm thanh 2D.

### 1.2.12 Camera

Là một game object đặc biệt trong scene, dùng để xác định tầm nhìn, quan sát các đối tượng khác trong game. Bao gồm các thuộc tính:

* Clear flags: Xác định phần nào của màn hình sẽ bị xóa. Đây là tiện dụng khi sử dụng nhiều máy ảnh để vẽ các yếu tố trò chơi khác nhau.
* Background: Màu áp dụng cho các màn hình còn lại sau khi tất cả các yếu tố trong quan điểm đã được rút ra và không có skybox.
* Culling Mask: Bao gồm hoặc bỏ qua lớp của các đối tượng được đưa ra bởi các Camera.
* Projection: khả năng của máy ảnh để mô phỏng góc nhìn.
* Field of view (thuộc tính chỉ xuất hiện khi chọn Perspective trong mục Projection): Chiều rộng của góc nhìn của Camera, đo bằng độ dọc theo trục Y.
* HDR: Cho phép High Dynamic Range dựng hình cho camera này.

### 1.2.13 Transform

* Transform: quản lý object trong không gian ba chiều, theo ba thông số:
* Position: quản lý vị trí hiện tại của object.
* Rotation: quản lý các thông số quay của object theo các trục x, y, z.
* Scale: quản lý các thông số phóng to, thu nhỏ theo các trục x, y, z.

### 1.2.14 Renderer

Các SpriteRenderer thành phần cho phép bạn hiển thị hình ảnh như Sprites để sử dụng trong cả hai cảnh 2D và 3D.

* Sprite: Các đối tượng Sprite để render. Đối tượng Sprite có thể được tạo ra từ textures bằng cách sử dụng các thiết lập Sprite.
* Color: Vertex màu.
* Material: Chất liệu được sử dụng để làm sprite.
* Sorting Layer: Các layer được sử dụng để xác định các ưu tiên của sprite này trong khi hiển thị.
* Order in Layer: Các ưu tiên của sprite trong layer của nó. Con số thấp hơn được kết xuất đầu tiên và con số tiếp theo phủ bên dưới.

## 1.3 TỔNG QUAN VỀ VISUAL STUDIO

### 1.3.1 Giới thiệu

Visual studio là một phần mềm hỗ trợ đắc lực hỗ trợ công việc lập trình website. Công cụ này được tạo lên và thuộc quyền sở hữu của ông

lớn công nghệ Microsoft. Năm 1997, phần mềm lập trình nay có tên mã Project Boston. Nhưng sau đó, Microsoft đã kết hợp các công cụ phát triển, đóng gói thành sản phẩm duy nhất.

Visual Studio là hệ thống tập hợp tất cả những gì liên quan tới phát triển ứng dụng, bao gồm trình chỉnh sửa mã, trình thiết kế, gỡ lỗi. Tức là, bạn có thể viết code, sửa lỗi, chỉnh sửa thiết kế ứng dụng dễ dàng chỉ với 1 phần mềm Visual Studio mà thôi. Không dừng lại ở đó, người dùng còn có thể thiết kế giao diện, trải nghiệm trong Visual Studio như khi phát triển ứng dụng Xamarin, UWP bằng XAML hay Blend vậy.

### 1.3.2 Tính năng

Tính đến nay, Visual Studio vẫn được coi là phần mềm lập trình hệ thống hàng đầu, chưa có phần mềm nào có thể thay thế được nó. Được đánh giá cao như vậy bởi Visual Studio sở hữu nhiều tính năng cực kỳ hấp dẫn. Cụ thể:

Đa nền tảng: Phần mềm lập trình Visual Studio của Microsoft hỗ trợ sử dụng trên nhiều nền tảng khác nhau. Không giống như các trình viết code khác, Visual Studio sử dụng được trên cả Windows, Linux và Mac Systems. Điều này cực kỳ tiện lợi cho lập trình viên trong quá trình ứng dụng.

Đa ngôn ngữ lập trình

* Không chỉ hỗ trợ đa nền tảng, Visual Studio cũng cho phép sử dụng nhiều ngôn ngữ lập trình khác nhau từ C#, F#, C/C++, HTML, CSS, Visual Basic, JavaScript,… Bởi vậy, Visual Studio có thể dễ dàng phát hiện và thông báo cho bạn khi các chương trình có lỗi.

Hỗ trợ website

Visual Studio code cũng hỗ trợ website, đặc biệt trong công việc soạn thảo và thiết kế web.

Kho tiện ích mở rộng phong phú: Mặc dù Visual Studio có hệ thống các ngôn ngữ hỗ trợ lập trình khá đa dạng. Nhưng nếu lập trình viên muốn sử dụng một ngôn ngữ khác, bạn có thể dễ dàng tải xuống các tiện ích mở rộng. Tính năng hấp dẫn này được hoạt động như một phần chương trình độc lập nên không lo làm giảm hiệu năng của phần mềm.

Lưu trữ phân cấp: Phần lớn các tệp dữ liệu đoạn mã của Visual Studio đều được đặt trong các thư mục tương tự nhau. Đồng thời, Visual Studio cũng cung cấp một số thư một cho các tệp đặc biệt để bạn lưu trữ an toàn, dễ tìm, dễ sử dụng hơn.

Kho lưu trữ an toàn: Với Visual Studio, bạn có thể hoàn toàn yên tâm về tính lưu trữ, bởi phần mềm đã được kết nối GIT và một số kho lưu trữ an toàn được sử dụng phổ biến hiện nay.

Màn hình đa nhiệm: Visual Studio sở hữu tính năng màn hình đa nhiệm, cho phép người dùng mở cùng lúc nhiều tập tin, thư mục dù chúng có thể không liên quan tới nhau.

Hỗ trợ viết code: Khi sử dụng code vào trong lập trình, với Visual Studio, công cụ này có thể đề xuất tới các lập trình viên một số tùy chọn thay thế nhằm điều chỉnh đôi chút để đoạn code áp dụng thuận tiện hơn cho người dùng.

Hỗ trợ thiết bị đầu cuối: Phần mềm Visual Studio cũng tích hợp các loại thiết bị đầu cuối, giúp người dùng không cần chuyển đổi giữa hai màn hình hay trở về thư mục gốc khi thực hiện một thao tác cần thiết nào đó.

Hỗ trợ Git: Do kết nối với GitHub nên Visual Studio cho phép hỗ trợ sao chép, kéo thả trực tiếp. Các mã code này sau đó cũng có thể thay đổi và lưu lại trên phần mềm.

Intellisense: Tính năng nhắc Intellisense được sử dụng hầu hết trong các phần mềm lập trình, bao gồm cả Visual Studio. Tuy nhiên, so với các trình viết mã, Visual Studio vẫn được đánh giá cao về tính chuyên nghiệp. Đặc biệt, tính năng này còn có thể phát hiện tất cả các đoạn mã không đầy đủ, nhắc lập trình viên, gợi ý sửa đổi, khai báo biến tự động trong trường hợp lập trình viên quên, giúp bổ sung cú pháp còn thiếu,…

Tính năng comment: Một tính năng cũng khá hay ho, hỗ trợ cho người lập trình trong trường hợp “nhớ nhớ quên quên” đó là tính năng bình luận. Tính năng này cho phép lập trình viên để lại nhận xét, giúp dễ dàng ghi nhớ công việc cần hoàn thành, không bỏ sót công đoạn nào.

# CHƯƠNG 2.PHÁT TRIỂN KỊCH BẢN VÀ THIẾT KẾ GAME

## 2.1. Giới thiệu tổng quan

### 2.1.1. Thông tin game

“Freight Transportation” là một trò chơi 2D thuộc thể loại nhập vai, phiêu lưu. Đồ họa được thiết kế đơn giản với đồ họa pixel, tạo cảm giác cổ điển nhưng thân thuộc cho người chơi. Được phát triển cho nền tảng thiết bị PC và trong tương lai có thể là các nền tảng khác

### 2.1.2. Thể loại game và yếu tố game

Đây là một tựa game 2D thuộc thể loại nhập vai.

Khái niệm game nhập vai

* Game nhập vai (Role-playing games), viết tắt là RPG là tựa game mà khi người chơi hay game thủ tham gia tựa game này, người chơi sẽ được hóa thân thành một nhân vật trong Game có khả năng điều khiển và kiểm soát nhân vật .

Khái niệm game phiêu lưu

Trò chơi phiêu lưu hay trò chơi mạo hiểm là một thể loại video game mà trong đó giả định người chơi là nhân vật chính trong một câu chuyện có tính tương tác tiến triển theo hướng khám phá và vượt qua thử thách. Game thuộc thể loại này thường xây dựng cốt truyện đủ lôi cuốn dễ giữ chân người chơi, cho phép nó được diễn dịch sang một phạm vi rộng lớn loại hình truyền thông mang tính tường thuật như sách vở hay điện ảnh, và bao hàm hàng loạt thể loại văn chương. Gần như toàn bộ trò chơi phiêu lưu (dưới hình thức văn bản hay đồ họa) được thiết kế cho một người chơi, bởi vì chúng nhấn mạnh vào kịch bản và nhân vật, khiến cho nhiều người chơi sẽ trở nên khó khăn.

Game mang lại nhiều yếu tố như :

* Tăng khả năng quan sát và sự nhanh nhẹn
* Mang tính giải trí.
* Cơ chế game đơn giản nhưng yêu cầu độ tập chung cao.

### 2.1.3. Đối tượng chơi

Trò chơi được phát triển cho những độ tuổi trẻ để giải trí, thư giãn sau những phút học tập, làm việc mệt mỏi. Với lối chơi game đơn giản bằng các thao tác bấm phím/chuột và luật chơi dễ hiểu. Độ tuổi tối thiểu để có thể thao tác game phù hợp là từ trên 12 tuổi.

### 2.1.4. Nền tảng

Game được xây dựng bằng Unity nên có khả năng phát triển tốt trên các nền tảng lớn. Nhưng hiện tại game tập trung chủ yếu vào nền tảng window (Laptop và Desktop)

## 2.2. Kịch bản game

### 2.2.1. Mô tả

Nhân vật chính trong game là một tài xế có biệt danh “John”, anh là một “quái xế” nổi tiếng và có những cuộc rượt đuổi thót tim trên đường phố. Trong một lần làm nhiệm vụ, anh vướng vào một vụ án thanh toán nhau giữa các băng cướp. Anh bị 1 trong 2 nhóm đe dọa dù đã biết có nhiều nguy hiểm nhưng anh vẫn nhận phải nhận nhiệm vụ giao hàng đến nơi chỉ định. Nhưng có vẻ như phía đối thủ đã nắm được lộ trình của anh, chúng tạo ra những cạm bẫy đẻ phá hoại nhiệm vụ của anh. Việc của anh là phải tập trung sống sót trên con đường vận chuyển và tìm ra kẻ nào hãm hại mình.

Người chơi điều khiển nhân vật chính vượt qua các chướng ngại vật. Nhặt các vật phẩm tăng tốc để có thể vượt qua các chướng ngại vật.

### 

### 

### 

### 2.2.2. Luật chơi

Game cho phép người chơi tự do di chuyển theo không gian game quy ước. Không gian quy là các vạch kẻ đường. Có thời gian quy định.

Điều kiện thắng:

* Người chơi đến điểm nhận hàng và nhặt hàng.
* Người chơi giao hàng đến điểm nhận tránh các chướng ngại vật và nhặt các item tăng chỉ số và tốc độ.

Điều kiện thua:

* Không đủ điểm trong thời gian quy định.

### 2.2.3. Tính điểm

Game tính điểm dựa trên số lượng gói hàng giao điểm đích: Số lượng hàng giao \* 10.

### 2.2.4. Tương tác và điều khiển game

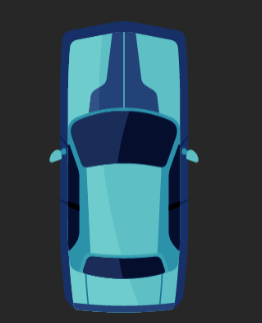
Tất cả các thao tác điều khiển trò chơi đều được thực hiện bởi bàn phím và chuột máy tính để di chuyển, thực hiện kỹ năng.

* Di chuyển sang trái: Phím mũi tên lên trên hoặc phím W.
* Di chuyển sang phải: Phím mũi tên xuống dưới hoặc phím S.
* Chuyển hướng xe sang trái: Phím mũi tên sang trái hoặc phím A;
* Chuyển hướng xe sang phải: Phím mũi tên sang phải hoặc phím D;

### 2.2.5. Các phần tử của game (Game Element)

#### 2.2.5.1. Player

Player: Chiếc xe Lamborghini của John:



Hình 2.1: Xe của nhân vật

#### 2.2.5.3. Enemy

Enemy: Là các chướng ngại vật, phương tiện qua lại.

| Đối tượng | Đặc điểm | Mô tả | Hình ảnh |
| --- | --- | --- | --- |
| Ô tô Hyundai | Có khả năng di chuyển | Xe chạy trên đường cao tốc | Hình 2.12: Hyundai |
| House 1 | Chướng ngại vật trên đường | nằm trên lề đường quốc lộ | Hình 2.12: House 1 |
| House 2 | Chướng ngại vật | nằm trên lề đường quốc lộ | Hình 2.13: House 2 |
| Tree 1 |  |  | Hình 2.14: Tree |
| Corner | Hình vòng cung | Xe va chạm vào phần rìa của làn đường sẽ bị tính là phạm lội |  |

*Bảng 2.2: Enemy list*

### 2.2.6. Thiết kế các lv của game

Game sẽ có tổng cộng 3 màn chơi, mỗi màn sẽ có yêu cầu về số lượng hàng cần giao tăng dần và map cũng lớn dần, cụ thể như sau:

* Level 1: Nhặt được 1 hộp hàng và giao đến nơi quy định.
* Level 2 : Nhặt được 5 hộp hàng và giao đến nơi quy định.
* Level 3 :Nhặt được 10 hộp hàng và giao đến nơi quy định.

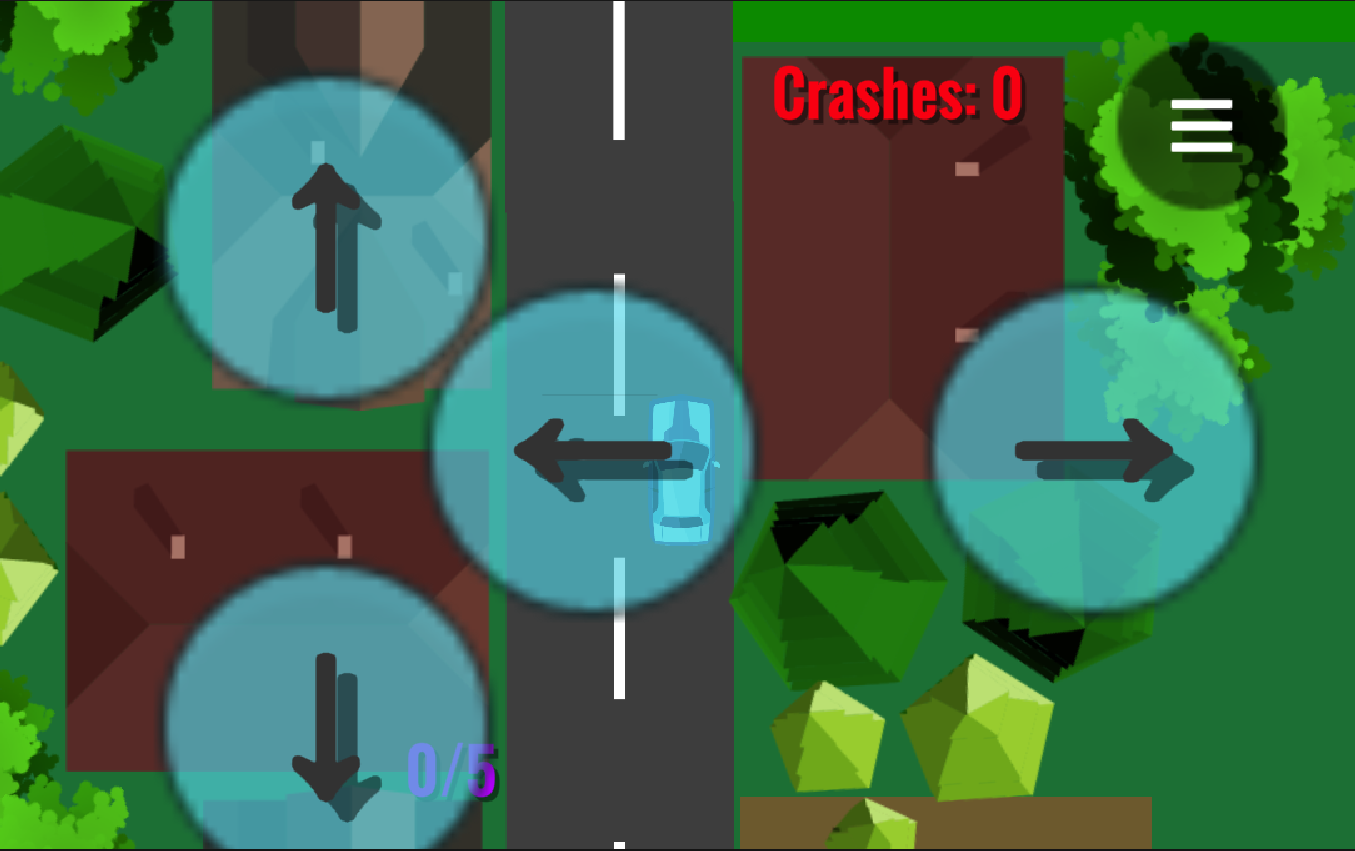
## 2.3 Thiết kế giao diện

### 2.3.1. Màn khởi đầu



Hình 2.16: Hình dung màn hình menu chính

### 2.3.2. Màn thứ hai



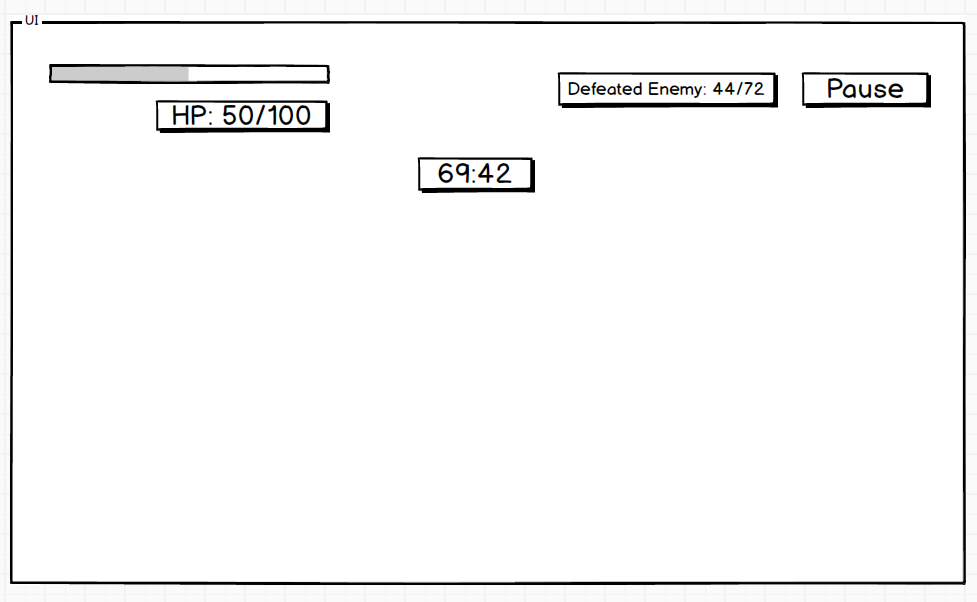
Hình 2.17: Màn hình thứ hai

### 2.3.3. Màn hình thứ ba



Hình 2.18: Màn hình thứ ba

### 2.3.4: UI trong game



Hình 2.19 Hình dung màn hình UI

### 2.3.5. Màn hình Pause

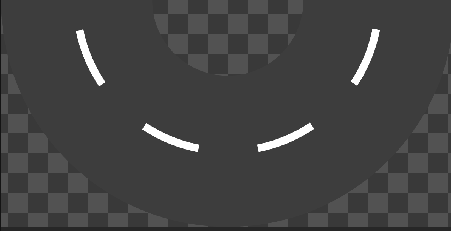


Hình 2.20: Hình dung màn hình Pause Menu

## 2.4. Tài nguyên:

### 2.4.1. Hình ảnh:

Background Sử dụng: <https://www.dfoneople.com/media/wallpaper>



Hình 2.24: Ví dụ hình ảnh

### 2.4.2. Âm thanh

* BGM: Dungeon Fighter Online BGM, Shin Megami Tensei 5 Vengeance OST, Shin Megami Tensei 4 OST, ICEY OST
* SFX: <https://www.sounds-resource.com/pc_computer/dnfduel/>

### 2.4.3. Game Assest

* Player:
* Enemy/Boss: <https://itch.io>
* World, Map asset: <https://brullov.itch.io/oak-woods>

### 2.4.4. Tham khảo tutorial:

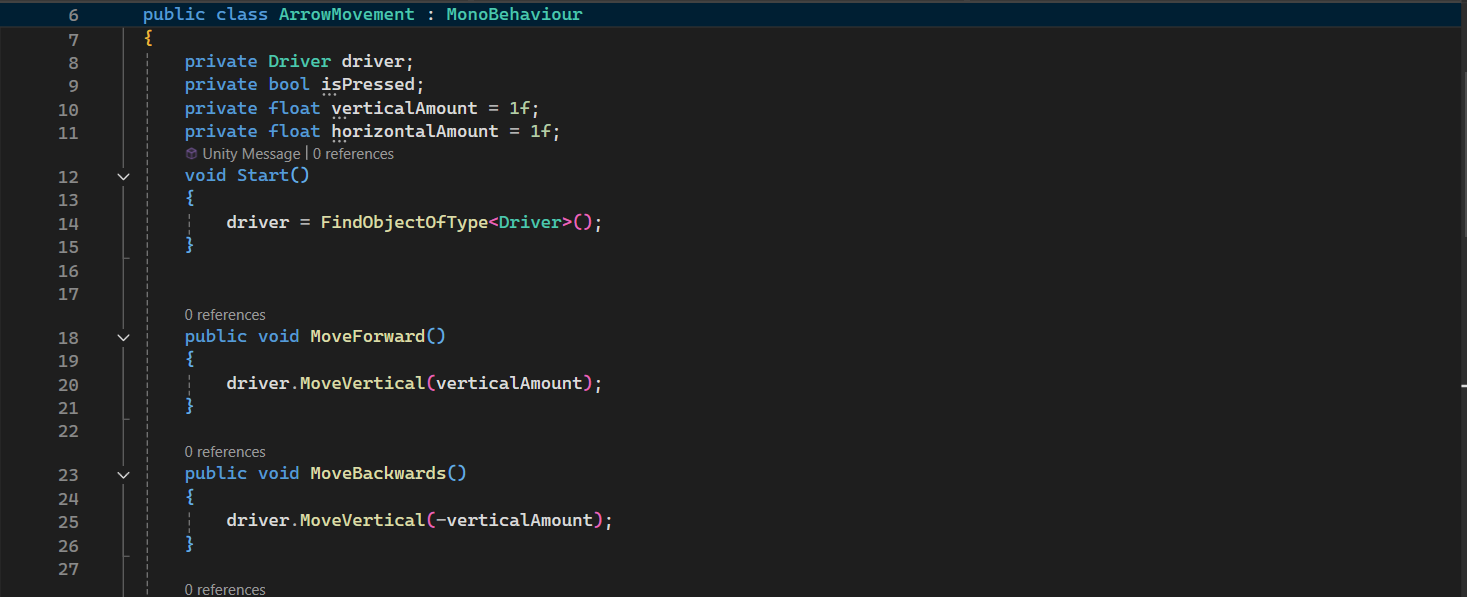
<https://www.youtube.com/@ChrisTutorialsYT>

# CHƯƠNG 3: CÁC KĨ THUẬT XÂY DỰNG GAME VÀ KẾT QUẢ THỰC NGHIỆM.

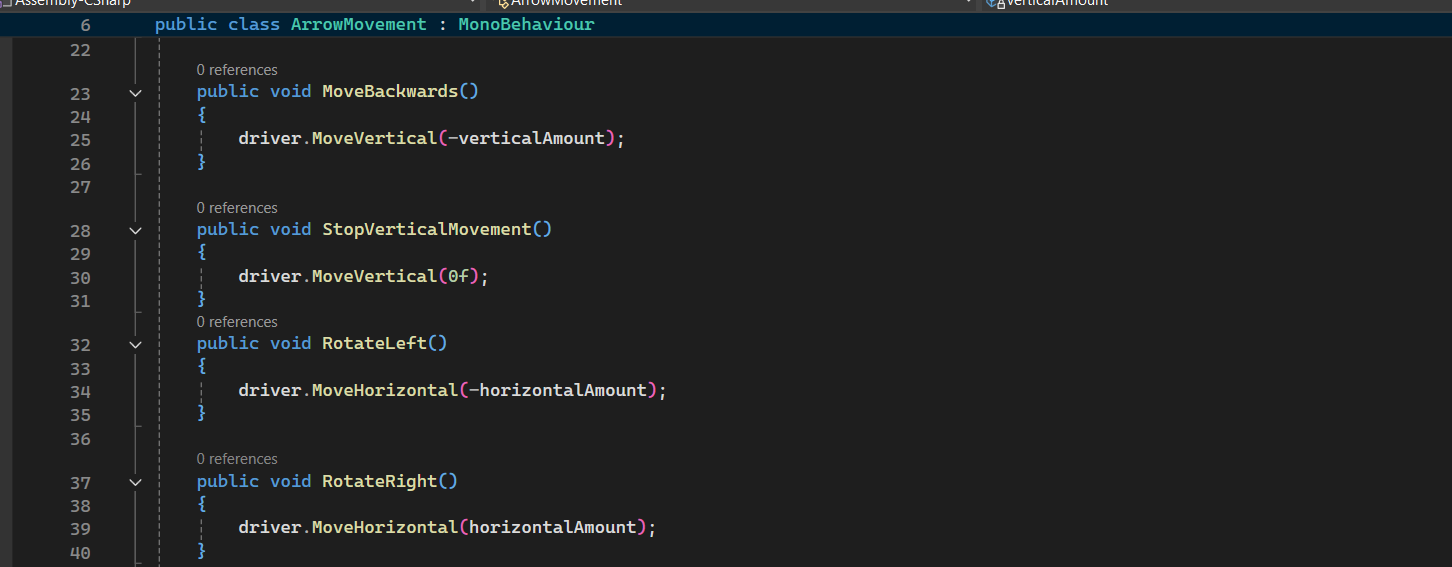
## 3.1. Các kĩ thuật xây dựng game

### 3.1.1 Các kĩ thuật với Player

#### 3.1.1.1 Di chuyển



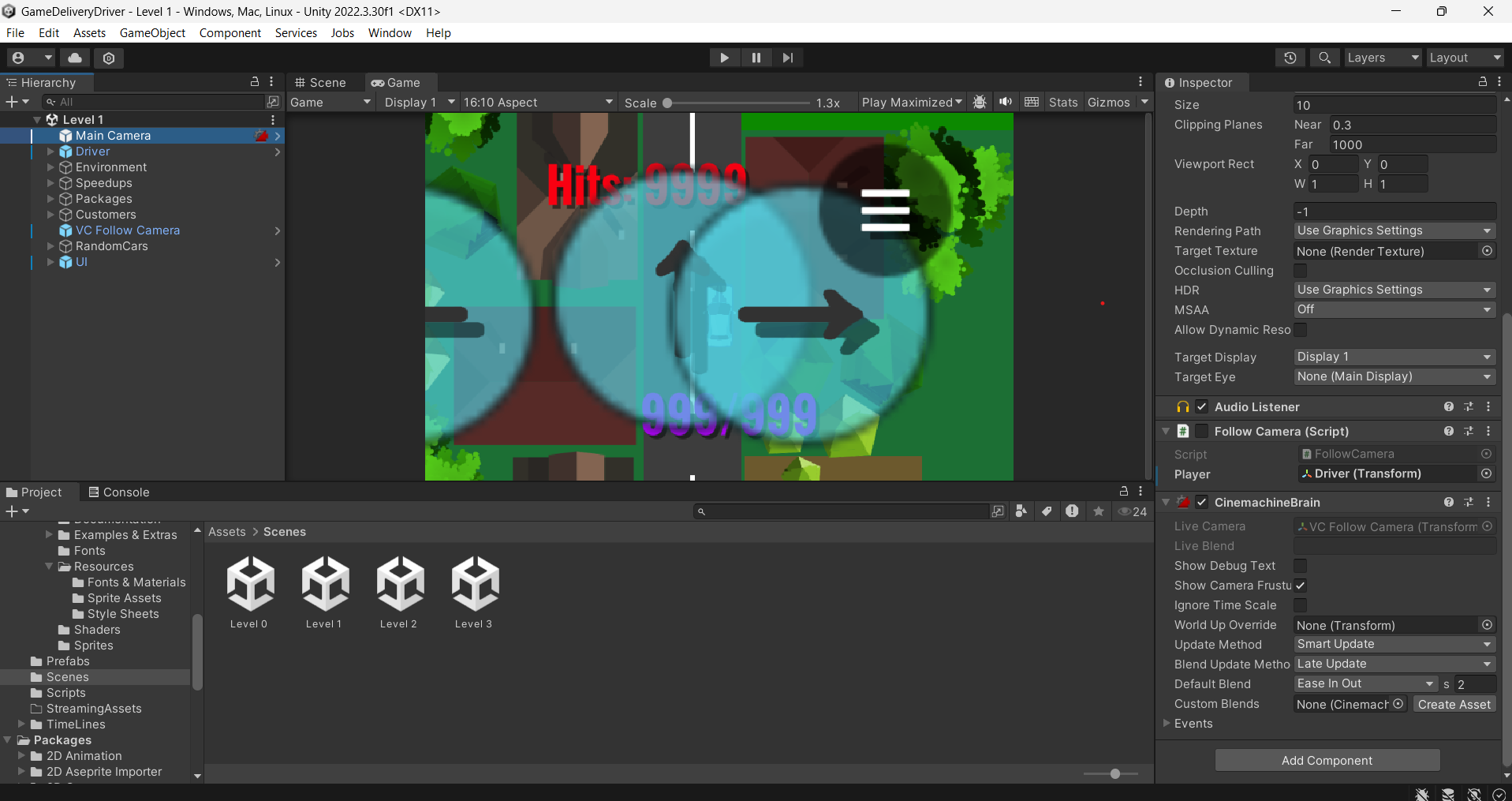
Hình 3.1: Code di chuyển 1



Hình 3.2: Code di chuyển 2

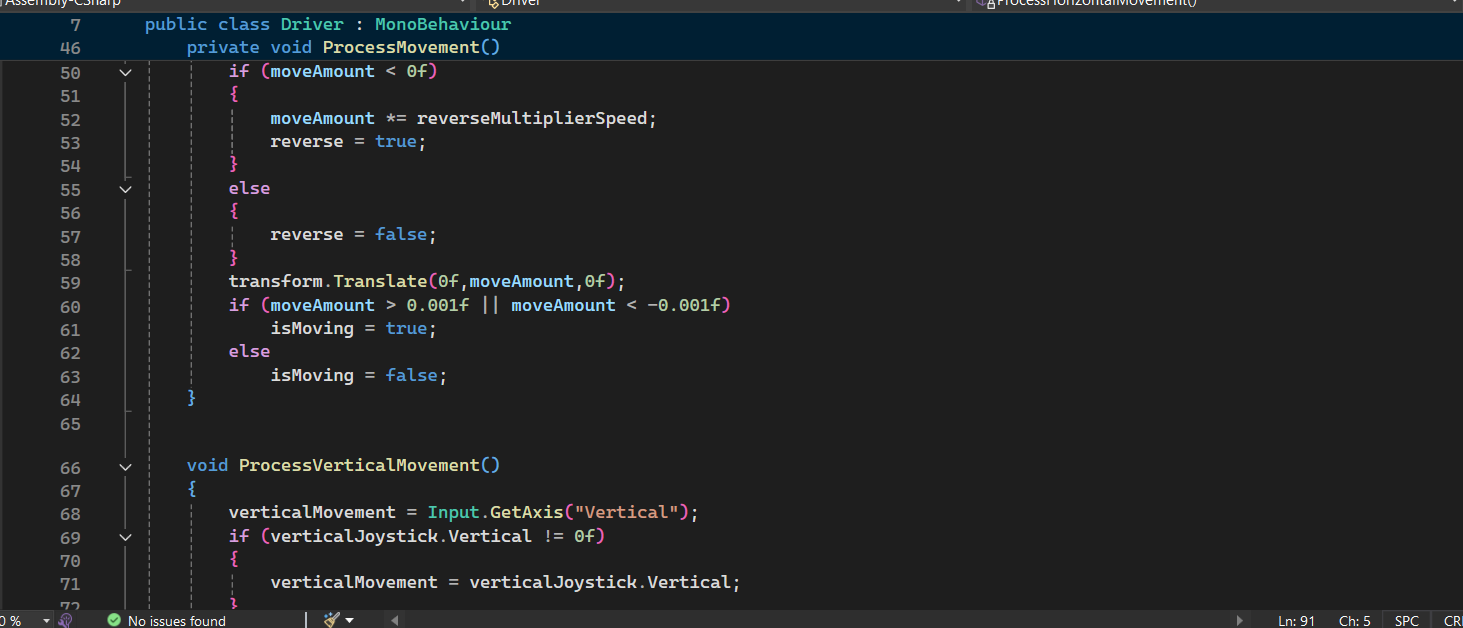
#### 3.1.1.2. Camera theo dõi player

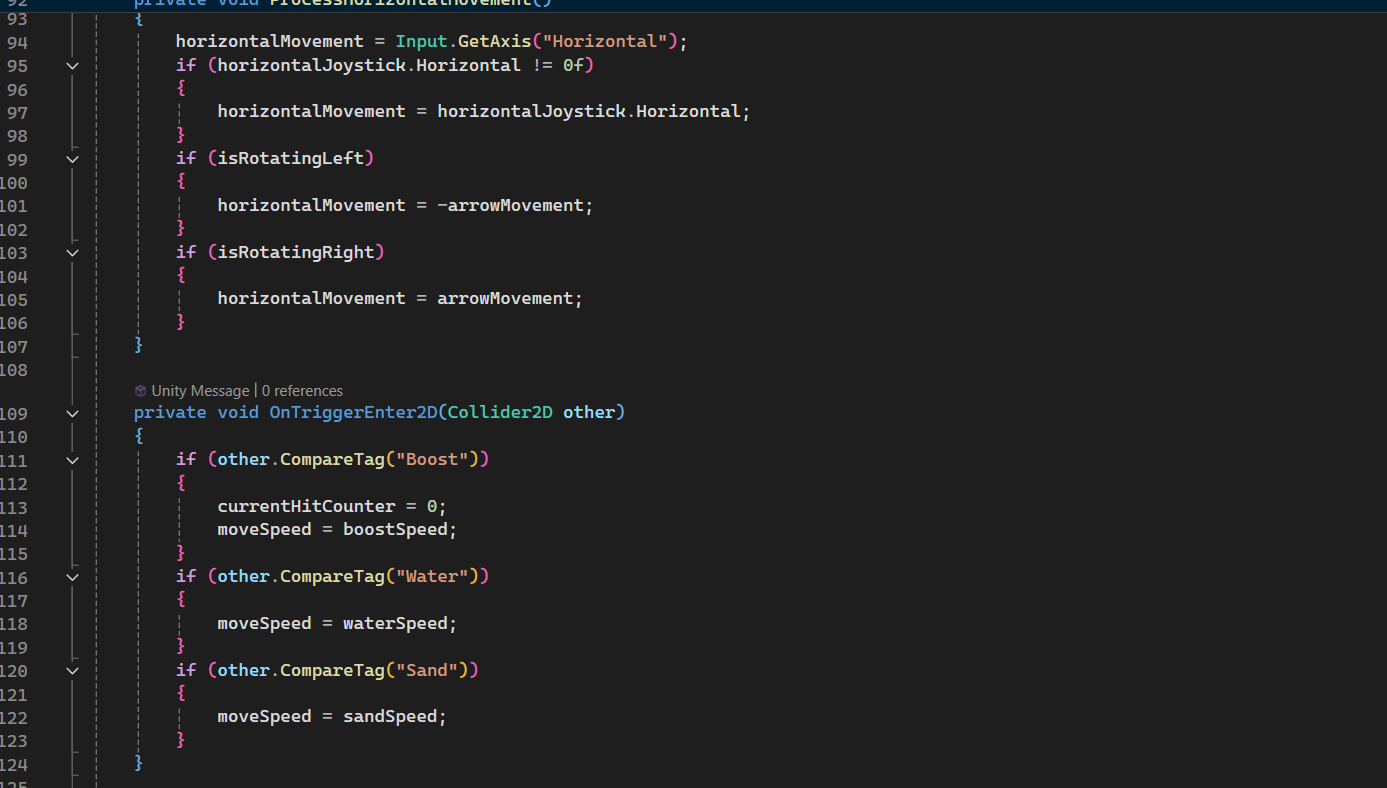
Sử dụng Cinemachine Brain để tracking player cũng như giới hạn camera trong một khoảng không chỉ định bởi Camera Boundary.



Hình 3.3: Cinemachine

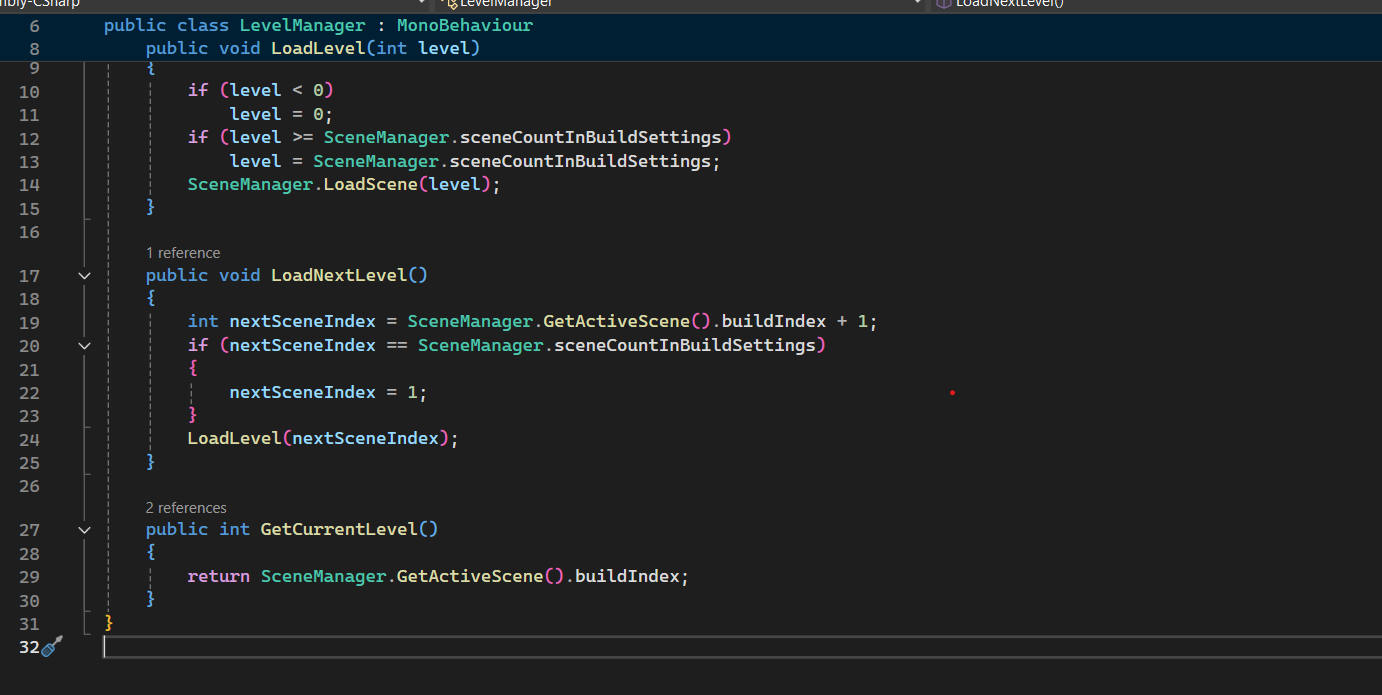
#### 3.1.1.3. Thay đổi tốc độ và di chuyển





Hình 3.4: Thay đổi tốc độ và hướng di chuyển

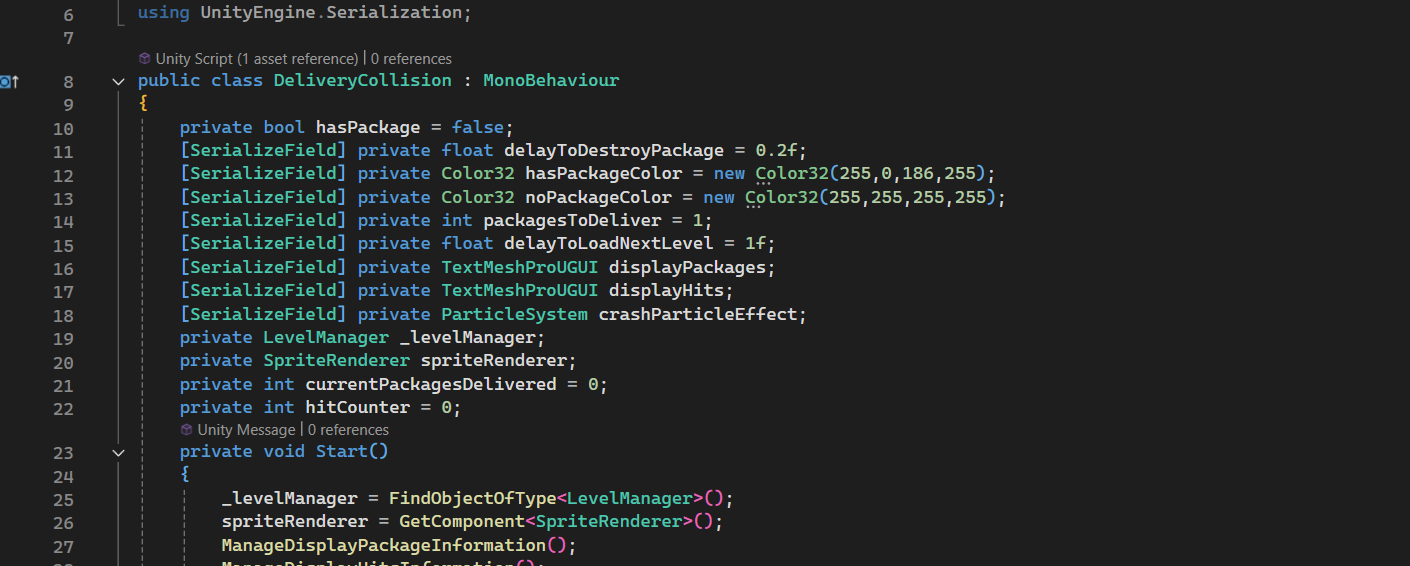
#### 3.1.1.4. Chuyển level khi thắng game

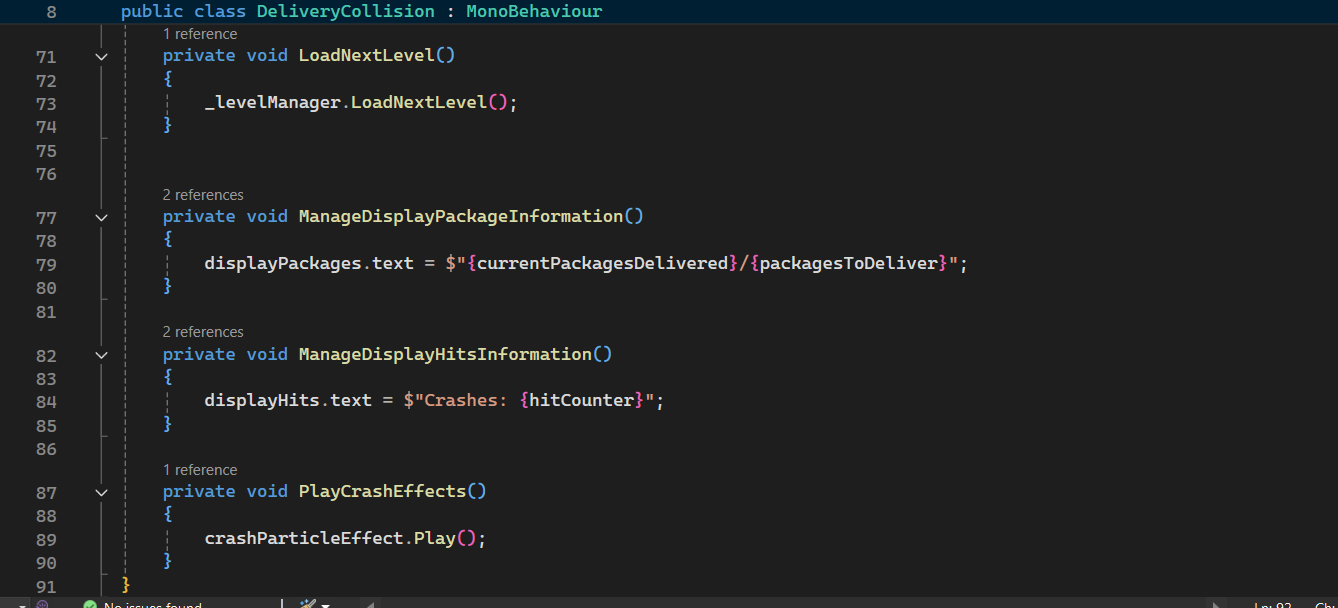
****

Hình 3.5: Chuyển level khi thắng game

#### 3.1.1.5. Quản lý và xử lý va chạm với các đối tượng khác

Khi player va chạm vào các chướng ngại vật thì crash sẽ tăng lên 1 điểm.

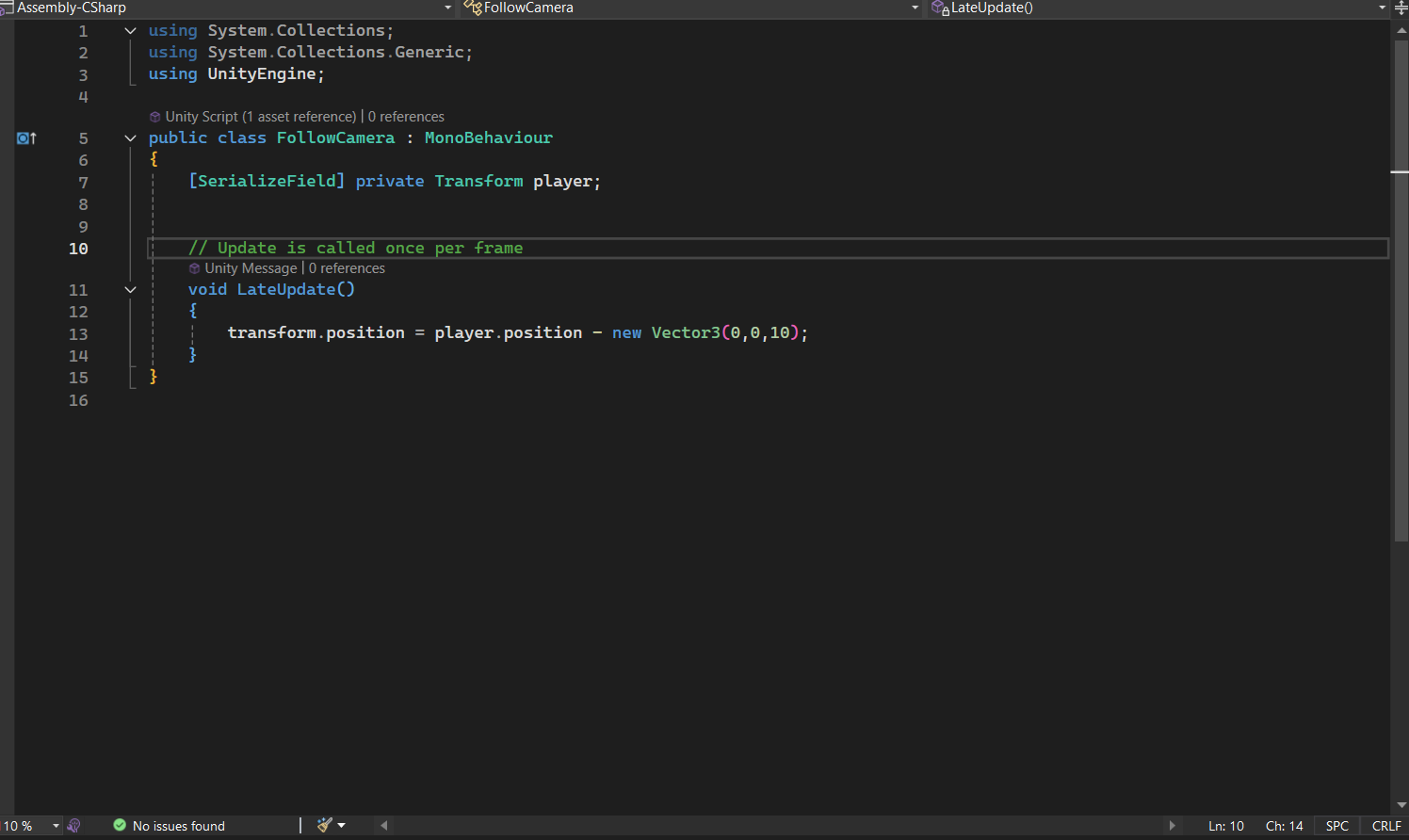




Hình 3.7: Xử lý va chạm collider của Hitbox

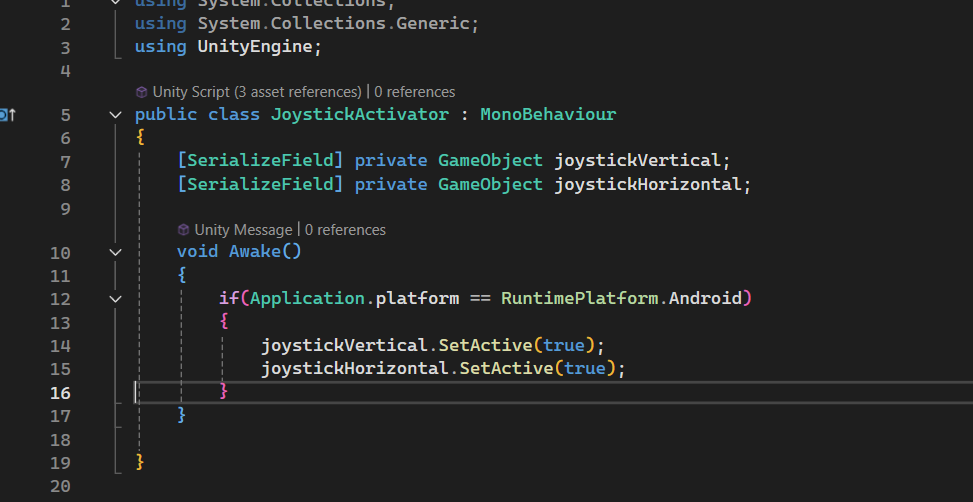
#### 3.1.2.1. Các hành vì của các đối tượng

- Follow Camera: Theo dõi camera theo player



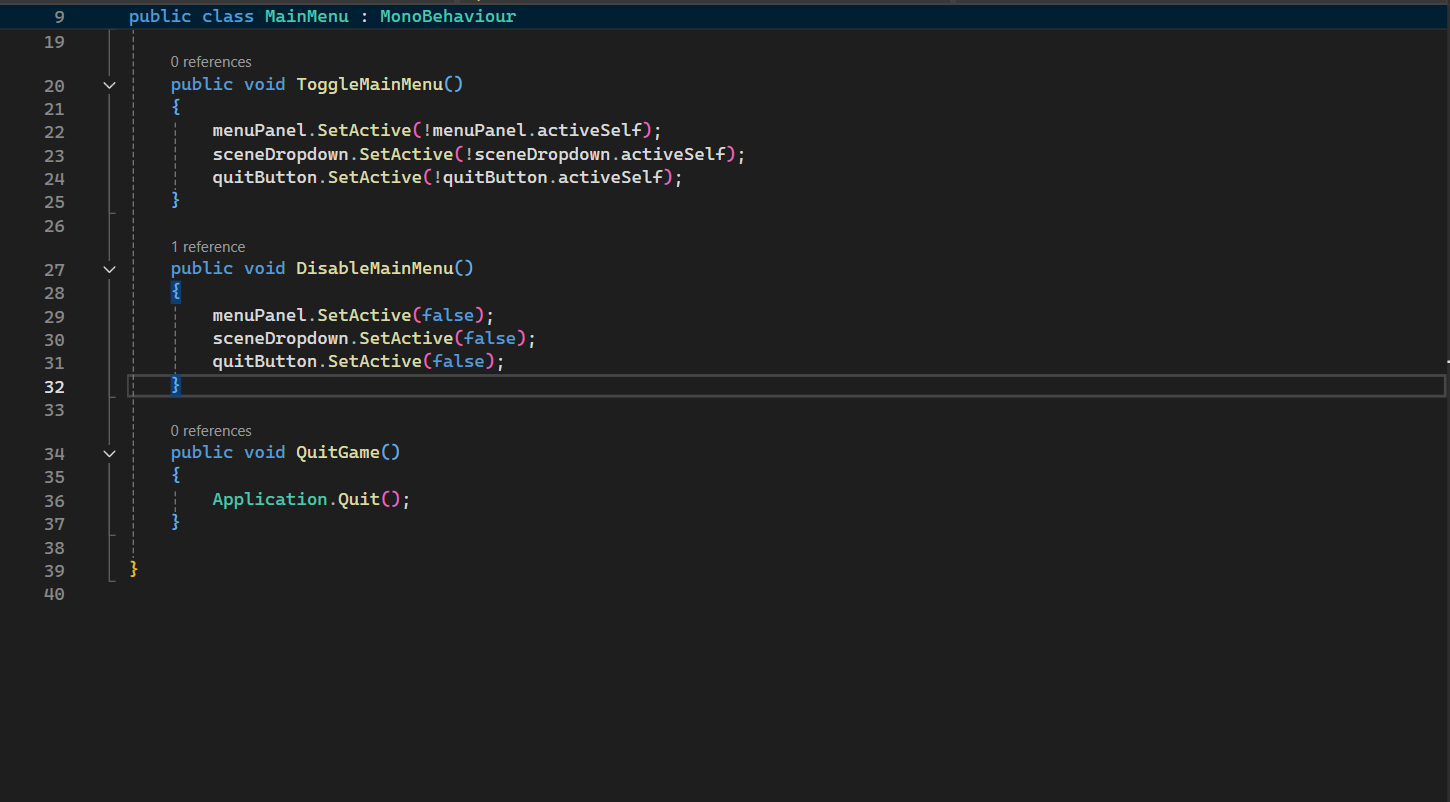
Hình 3.8: Player Tracking Scripts

-Tạo ra JoyStick tạo ra các thanh điều hướng có thể chạy trên nền tảng android.



Hình 3.8:JoyStick chạy trên nền tảng android.

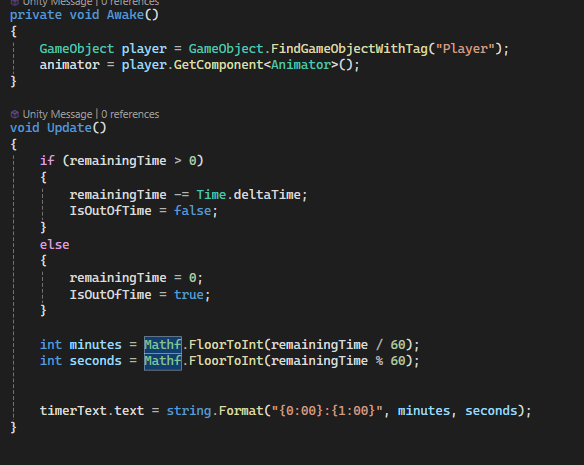
#### 3.1.2.2. Các thành phần khác:

Tạo ra main menu để có thể quit nếu thấy game khó 

Hình 3.10: Tạo ra Main menu

### 3.1.3. Xây dựng timer.

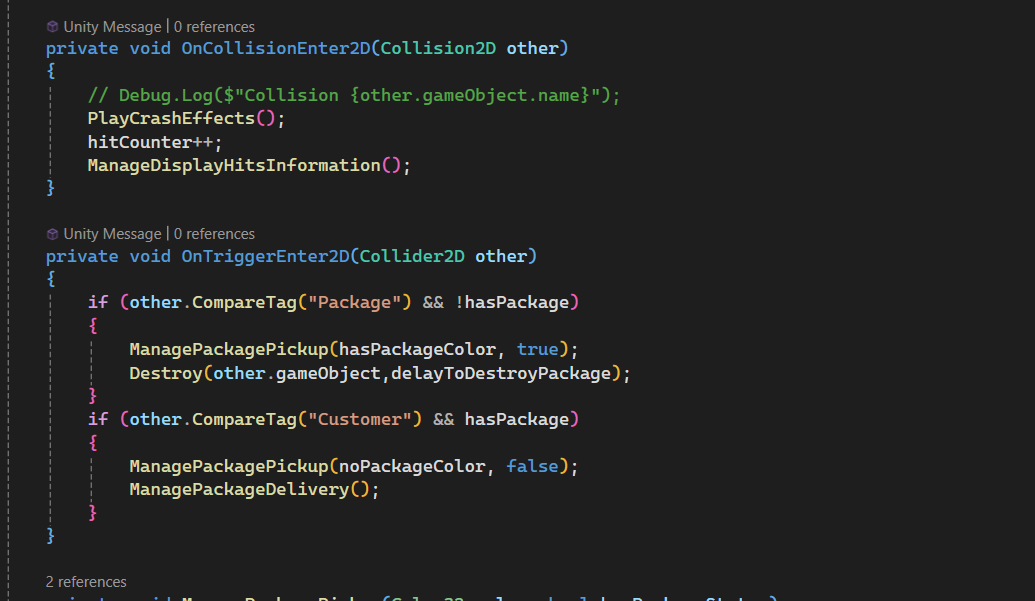
Xây dựng timer và bộ đếm lùi bằng cách trừ đi delta timer và format lại theo dạng int, nếu hết giờ thì sẽ ánh xạ lên “Player” là đã hết thời gian.



Hình 3.15: Timer

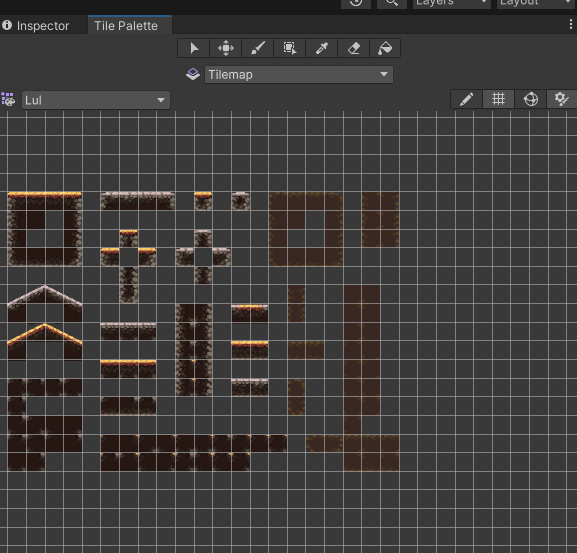
### 3.1.4. Kỹ thuật đếm các các gói hàng mà nhân vật giao được, số hit nhân vật đã va chạm với các chướng ngại vât

Khởi tạo các biến đếm kẻ địch nhận vào cùng với điều kiện chiến thắng ở static, mục đích là để gọi lại ở mọi nơi, mỗi khi bật stage lên tất cả các chỉ số đều được reset. Nếu hoàn thành điều kiện thì điều kiện chiến thắng sẽ được thỏa mãn.



Hình 3.16: Khởi tạo bộ đếm + condition

3.1.6 Kỹ thuật thiết kế map

Sử dụng Tilemap

Hình 3.20: Tile map

## 3.2 Kết quả thực nghiệm

Qua các kĩ thuật và xây dựng thiết kế ở trên, chúng ta thu được các màn hình dưới đây:

### 3.2.1. Main menu



Hình 3.21: Level 1



Hình 3.21:Level 2



Hình 3.22: Level 3



Hình 3.23: Main menu

# KẾT LUẬN

**Kết quả đạt được**

Từ chương đầu, chúng em em đã trình bày được cơ sở lý thuyết, công cụ để thực hiện và vận dụng kiến thức cơ sở học được ở trường giúp giải quyết đề tài này. Cải thiện việc tự học, tự nghiên cứu cách xây dựng, thiết kế game và lập trình game 2D trong Unity. Qua đó thành thạo việc tạo và sử dụng các hoạt ảnh animation, chỉnh sửa animator. Trong sản phẩm game thì: Player và Enemy có đa dạng kỹ năng tấn công; giao diện menu bắt mắt, âm thanh game sống động, cuối màn game sẽ đánh giá xếp loại điểm, tổng kết màn chơi.

**Chưa đạt được**

Vì thời gian và nguồn tài nguyên có hạn, chúng em có một vài thứ chưa đạt được mong muốn như: hiệu ứng UI còn hạn chế; số lượng màn chơi không nhiều, chỉ có 3 màn; số lượng kẻ địch, cơ chế không nhiều như mong đợi; vẫn chưa có những kẻ địch có gimmick và cách khắc chế đặc biệt; Platform còn hơi một màu và đơn điệu, cơ chế di chuyển và độ “thỏa” trong gameplay hack and slash chưa đặt như ban đầu cầu đặt ra.

**Thuận lợi**

Để xây dựng sản phẩm này, em rất may mắn khi có kiến thức cơ sở được học ở trường về thiết kế game trên Unity và được giáo viên chỉ bảo và bạn bè giúp đỡ nhiệt tình. Ngoài ra thì tài liệu trên internet rất nhiều, thoải mái nghiên cứu, tìm hiểu cách xây dựng, thiết kế game và lập trình game 2D trong thời gian làm bài tập lớn.

**Khó khăn**

Sự khó khăn và thiếu thốn trong asset là rào cản lớn nhất trong quá trình làm nên”Freight Transportation”, bởi vì tiềm năng asset không cho phép, chúng em không thể làm cho player lung linh và có nhiều cơ chế đặc biệt được. Giới hạn về thời gian cũng là một vấn đề rất lớn trong quá trình hoàn thiện và phát triển bài tập lớn.

**Kinh nghiệm rút ra**

Biết cách sắp xếp thời gian hợp lý nếu không sẽ gặp rất nhiều khó khăn. Hiểu được việc tự học và nghiên cứu là vô cùng quan trọng. Nắm bắt được cách chỉnh sửa hoạt ảnh, thao tác trên phần mềm Unity nhanh hơn.

**Hướng phát triển**

Tối ưu hoá, nâng cấp trò chơi: thêm chức năng chọn nhiều nhân vật, các nhân vật có những cơ chế khác biệt, thêm thời gian quy ước cho từng ván chơi.

# TÀI LIỆU THAM KHẢO

[1]. Tài liệu thầy Nguyễn Văn Tỉnh gửi.

[2]. Gameplay: <https://www.youtube.com/@ChrisTutorialsYT>

[3]. Github unity: <https://www.youtube.com/@Brackeys>

[4]. UI Design: <https://www.youtube.com/@BMoDev>

[5]. <https://www.reddit.com/r/unity/>