

BỘ GIÁO DỤC VÀ ĐÀO TẠO BỘ NÔNG NGHIỆP VÀ PTNT
TRƯỜNG ĐẠI HỌC THỦY LỢI



NGUYỄN THẾ VIỆT

TRÒ CHƠI CROSSY ROAD SỬ DỤNG UNITY

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, 2024

BỘ GIÁO DỤC VÀ ĐÀO TẠO BỘ NÔNG NGHIỆP VÀ PTNT
TRƯỜNG ĐẠI HỌC THỦY LỢI

NGUYỄN THẾ VIỆT

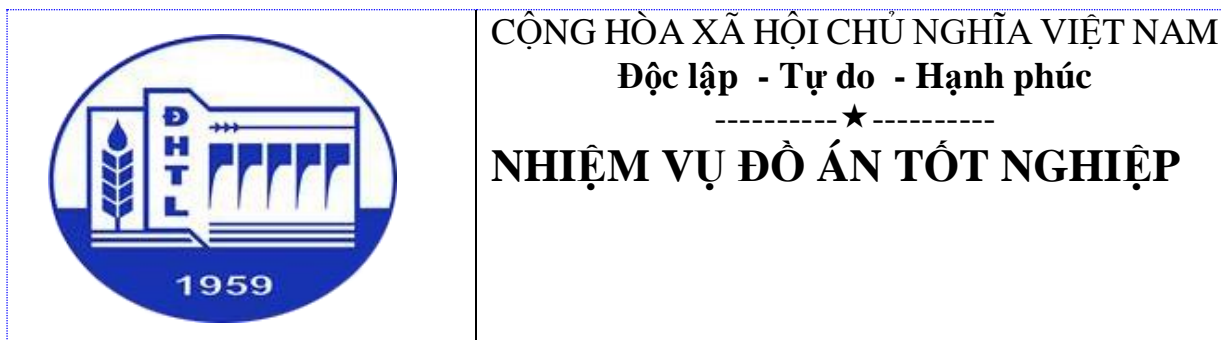
TRÒ CHƠI CROSSY ROAD SỬ DỤNG UNITY

Ngành: Công nghệ thông tin

Mã số: 7480201

NGƯỜI HƯỚNG DẪN: ThS. Trương Xuân nam

Hà Nội, 2024



Họ tên sinh viên: Nguyễn Thế Việt

Hệ đào tạo : Đại học chính quy

Lớp: 60TH3

Ngành: Công nghệ thông tin

Khoa: Công nghệ thông tin

- 1 TÊN ĐỀ TÀI:
TRÒ CHƠI CROSSY ROAD SỬ DỤNG UNITY
- 2 CÁC TÀI LIỆU CƠ BẢN:

- * Quy trình giao đồ án tốt nghiệp khoa Công nghệ thông tin Đại học Thủy Lợi (2021)
- * Website
- * https://vi.wikipedia.org/wiki/Crossy_Road
- * <https://docs.unity3d.com/Manual/index.html>

3. NỘI DUNG CÁC PHẦN THUYẾT MINH VÀ TÍNH TOÁN:

Nội dung cần thuyết minh	Tỷ lệ
1. Chương I. Tổng quan về ngành game	5%
2. Chương II. Tổng quan về Unity	31%
3. Chương III. Game Cross Road sử dụng Unity	11%
4. Chương IV: Phân tích và thiết kế hệ thống	40%
5. Chương V: Xây dựng và demo	12%
6. Chương VI: Kết luận	1%

4. GIÁO VIÊN HƯỚNG DẪN TỪNG PHẦN

Các phần	GVHD
1. Chương I. Tổng quan về ngành game	ThS. Trương Xuân Nam
2. Chương II. Tổng quan về Unity	
3. Chương III. Game Cross Road sử dụng Unity	
4. Chương IV: Phân tích thiết kế hệ thống	
5. Chương V: Xây dựng và demo	
6. Chương VI: Kết luận	

5. NGÀY GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Ngày 15 tháng 10 năm 2023

Trưởng Bộ môn
(Ký và ghi rõ Họ tên)

Giáo viên hướng dẫn chính
(Ký và ghi rõ Họ tên)

Nhiệm vụ Đồ án tốt nghiệp đã được Hội đồng thi tốt nghiệp của Khoa thông qua.

Ngày 15 tháng 10 năm 2023

Chủ tịch Hội đồng
(Ký và ghi rõ Họ tên)

Sinh viên đã hoàn thành và nộp bản Đồ án tốt nghiệp cho Hội đồng thi ngày 22 tháng 1 năm 2024

Sinh viên làm Đồ án tốt nghiệp

(Ký và ghi rõ Họ tên)

Nguyễn Thế Việt



TRƯỜNG ĐẠI HỌC THỦY LỢI
KHOA CÔNG NGHỆ THÔNG TIN

BẢN TÓM TẮT ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP

TÊN ĐỀ TÀI: Trò chơi Crossy Road sử dụng Unity

Sinh viên thực hiện: Nguyễn Thế Việt

Lớp: 60TH3

Mã sinh viên : 1851061396

Giáo viên hướng dẫn: ThS. Trương Xuân Nam.

TÓM TẮT ĐỀ TÀI

Game mobile ngày nay đang chiếm lĩnh thị trường giải trí với sự thuận tiện và mang lại trải nghiệm giải trí trên đầu ngón tay. Trong số các thể loại game, thể loại hyper casual đang là xu hướng nổi bật, dễ chơi, phù hợp cho mọi đối tượng người chơi.

Thể loại hyper casual đặc trưng bởi độ dễ dàng và tính chất giải trí ngắn gọn. Các tựa game trong thể loại này thường có gameplay đơn giản, đồ họa sáng tạo và khám phá các cơ chế chơi mới. Điều này giúp thu hút người chơi mới và giữ chân họ trong thời gian ngắn.

Đề tài được xây dựng dựa theo một trong những đại diện nổi bật của thể loại hyper casual là Crossy Road. Được phát triển bởi Hipster Whale. Trò chơi này không chỉ là một phiên bản hiện đại của trò chơi vượt đường cô điển mà còn mang lại trải nghiệm giải trí độc đáo và sáng tạo.

Với góc nhìn top-down, đồ họa voxel đẹp mắt, tạo nên một thế giới rực rỡ và đầy màu sắc. Người chơi sẽ điều khiển chú gà bang qua những con đường và sông để đạt điểm cao nhất có thể.

Gameplay của Crossy Road đơn giản nhưng gây nghiện với việc chạm vượt để điều khiển nhân vật, đồng thời phải quan sát và phản ứng nhanh để tránh chướng ngại vật không ngừng xuất hiện. Cùng với âm nhạc vui nhộn và âm thanh sinh động, trò chơi tạo ra một trải nghiệm giải trí độc đáo, phù hợp cho cả người chơi casual và hardcore. Crossy Road

không chỉ là một trò chơi, mà là một hành trình đầy thú vị, đem lại cho người chơi những giây phút giải trí đáng nhớ trên thiết bị di động của họ.

Tóm lại, Crossy Road là một trò chơi cực kỳ hấp dẫn và không thể bỏ qua đối với mọi người yêu thích game mobile.

Với những đặc điểm nêu trên, đó là lý do em chọn đề tài này để phát triển.

CÔNG NGHỆ SỬ DỤNG

- Unity
- C#

CÁC MỤC TIÊU CHÍNH

Mục tiêu:

- * Điều khiển nhân vật nhân vật bằng cách nhấn các phím W,A,S,D hoặc vuốt màn hình
- * Camera follow nhân vật
- * Map sinh ngẫu nhiên từ các mảnh địa hình cơ bản đường rừng, đường xe, đường ray xe lửa và đường sông. Và map sẽ sinh theo tiến trình di chuyển của người chơi
- * Chướng ngại vật sinh ngẫu nhiên
- * Tính điểm
- * Điểm cao nhất
- * Multi ending
 - o Nhân vật va chạm với phương tiện
 - o Nhân vật bị rơi xuống nước
 - o Nhân vật di chuyển ra khỏi khu vực cho phép của map
 - o Nhân vật không tiến lên trong 1 khoảng thời gian lâu hơn thời gian cho phép
- * Hệ thống gold, kiếm được trong quá trình chơi
- * Hệ thống nhân vật, dùng gold để mở khóa

Kỹ năng

- * Sinh viên tìm hiểu kiến thức về Unity, C# và các chức năng của Unity.
- * Sinh viên xây dựng thành công một game cơ bản với các chức năng đơn giản.

KẾT QUẢ DỰ KIẾN

Kiến thức

- * Sinh viên hiểu được về những kiến thức lập trình Unity, C#.
- * Sinh viên nắm được quy trình xây dựng một game trên Unity.

Kỹ năng

- * Báo cáo tổng hợp về Unity.
- * Thu thập yêu cầu, nghiệp vụ bài toán để đưa ra các phương án giải quyết phù hợp.
- * Sinh viên lập trình thành công game dựa trên nghiệp vụ bài toán bằng Unity viết bằng ngôn ngữ lập trình C# với bài toán chuyển phát nhanh.
- * Xây dựng một trò chơi hợp lý và thú vị.

LỜI CAM ĐOAN

Tác giả xin cam đoan đây là Đồ án tốt nghiệp của bản thân tác giả. Các kết quả trong Đồ án tốt nghiệp này là trung thực, và không sao chép từ bất kỳ một nguồn nào và dưới bất kỳ hình thức nào. Việc tham khảo các nguồn tài liệu (nếu có) đã được thực hiện trích dẫn và ghi nguồn tài liệu tham khảo đúng quy định.

Tác giả ĐATN

NGUYỄN THẾ VIỆT

LỜI CẢM ƠN

Trong suốt thời gian học tập rèn luyện tại trường Đại học Thủy Lợi, em xin gửi lời cảm ơn đến tập thể các thầy cô trong trường nói chung và các thầy cô khoa Công Nghệ Thông Tin nói riêng. Nhờ sự dìu dắt tận tình của các thầy cô mà em mới có đủ khả năng, kiến thức để làm Đồ án tốt nghiệp lần này cũng như là kinh nghiệm để bước ra ngoài làm việc.

Em xin gửi lời cảm ơn cảm ơn sâu sắc tới thầy TS. Trương Xuân Nam, ngành Công nghệ Phần mềm thuộc khoa Công nghệ Thông tin của trường Đại học Thủy Lợi. Trong quá trình thực hiện đồ án tốt nghiệp đã đưa cho em những lời khuyên, khuyến khích, góp ý tận tình để em hoàn thành đồ án của mình một cách tốt nhất. Đồ án này em nghĩ sẽ không thể hoàn thành nếu không có sự hỗ trợ về kiến thức, tài liệu và hướng đi của thầy.

Cuối cùng nhưng cũng rất quan trọng, em xin chân thành cảm ơn gia đình, người thân, bạn bè và những người đồng nghiệp, những người đã luôn bên cạnh em trong suốt thời gian sinh viên của mình tại trường Đại học Thủy Lợi.

MỤC LỤC

LỜI CAM ĐOAN	8
LỜI CẢM ƠN.....	9
MỤC LỤC	10
DANH MỤC HÌNH ẢNH.....	12
DANH MỤC BẢNG.....	13
CHƯƠNG 1: TỔNG QUAN VỀ NGÀNH GAME	14
1.1. Lịch sử	14
1.2. Các công cụ làm game.....	15
1.3. Các ngôn ngữ làm game	16
CHƯƠNG 2: TỔNG QUAN VỀ CÔNG CỤ LÀM GAME UNITY	18
2.1 Unity là gì?.....	18
2.2 Tính năng nổi bật của Unity	18
2.3.Các thành phần cơ bản trong Unity Editor	20
2.3.1 Scene View	20
2.3.2 Game View	21
2.3.3 Hierarchy Window	22
2.3.4 Project Window.....	23
2.3.5. Inspector Window	23
2.3.6. Console Window	24
2.4. Một số khái niệm cơ bản trong unity.....	25
2.4.1. GameObject	25
2.4.2. Prefabs	25
2.4.3. Assets.....	25
2.4.4. Sprite.....	25
2.4.5. Shader và Material.....	25
2.4.6. Camera.....	26
2.4.7. Component	26
2.4.8. Script.....	26
2.4.9. Transform.....	26
2.4.10. Collider	26
2.4.11. Rigidbody.....	26
2.5. Các hàm API phổ biến thường được sử dụng trong Unity	27

2.5.1. MonoBehaviour.....	27
2.5.2. GameObject	29
2.5.3. Transform.....	29
2.5.4. Input.....	29
2.5.5. Physics	29
2.5.6. Colliders.....	29
2.5.7. Audio	29
2.5.8. SceneManager	29
2.5.9. Time.....	30
2.5.10. Coroutine.....	30
2.5.11. ScriptableObject	30
CHƯƠNG 3: GAME CROSSY ROAD SỬ DỤNG UNITY.....	31
3.1 Phân tích thiết kế game.....	31
3.1.1. Đồ họa.....	31
3.1.2. Âm thanh.....	31
3.1.3. Giao diện.....	31
3.1.4. Điều khiển.....	31
3.1.5. Nhiệm vụ:.....	31
3.1.6. Map.....	31
3.1.6. Góc nhìn.....	32
3.1.7. Hệ thống tiền tệ (gold) và skin nhân vật.....	32
CHƯƠNG 4: PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	34
4.1. Xác định tác nhân.....	34
4.2. Xây dựng biểu đồ Usecase.....	34
4.3 . Danh sách use case:	34
4.4 . Đặc tả UseCase.....	35
4.4.1 Use case bắt đầu chơi.....	35
4.4.2 Use case điều khiển nhân vật:.....	36
4.4.3 Use case mở bảng nhân vật:	39
4.4.4 Use case mở khóa nhân vật:	41
4.4.5 Use case chọn nhân vật:	43
4.5 Dữ liệu	45
CHƯƠNG 5: XÂY DỰNG GAME VÀ DEMO	48
CHƯƠNG 6: KẾT LUẬN.....	53

TÀI LIỆU THAM KHẢO	54
--------------------------	----

DANH MỤC HÌNH ẢNH

Hình 2. 1 Logo Unity	18
Hình 2. 2 Các nền tảng hỗ trợ Unity	20
Hình 2. 3 : Scene View	21
Hình 2. 4 : Game View	22
Hình 2. 5 : Hierarchy Window	22
Hình 2. 6 : Project Window	23
Hình 2. 7: Inspector Window	24
Hình 2. 8 : Console Window.....	24
Hình 2. 9: Thứ tự thực thi trong Unity	28
Hình 4. 1: Mô hình UseCase tổng quát	34
Hình 4. 2: Sơ đồ Activity Use Case bắt đầu chơi	35
Hình 4. 3: Sơ đồ tuần tự Use Case bắt đầu chơi	36
Hình 4. 4: Sơ đồ Activity Use Case điều khiển nhân vật	37
Hình 4. 5: Sơ đồ tuần tự Use Case điều khiển nhân vật	38
Hình 4. 6: Sơ đồ Activity Use Case mở bảng nhân vật.....	40
Hình 4. 7: Sơ đồ tuần tự Use Case mở bảng nhân vật.....	41
Hình 4. 8: Sơ đồ Activity Use Case mở khóa nhân vật.....	42
Hình 4. 9: Sơ đồ tuần tự Use Case mở khóa nhân vật.....	43
Hình 4. 10: Sơ đồ Activity Use Case chọn nhân vật.....	44
Hình 4. 11: Sơ đồ tuần tự Use Case chọn nhân vật.....	45
Hình 4. 12: ScriptableObject.....	47
Hình 5. 1 Màn hình chính	48
Hình 5. 2 Màn hình game play.....	49
Hình 5. 3: Màn hình chọn nhân vật.....	50
Hình 5. 4 Màn hình khi người chơi kết thúc game với điểm cao nhất.....	51
Hình 5. 5 Một số hình ảnh trong game	52

DANH MỤC BẢNG

Bảng 4. 1: Kịch bản Use Case bắt đầu chơi.....	35
Bảng 4. 2: Kịch bản Use Case điều khiển nhân vật	36
Bảng 4. 3: Kịch bản mở bảng nhân vật	39
Bảng 4. 4: Kịch bản mở bảng khóa nhân vật	41
Bảng 4. 5: Kịch bản chọn nhân vật	43

CHƯƠNG 1: TỔNG QUAN VỀ NGÀNH GAME

1.1. Lịch sử

Ngành game đã trải qua một hành trình phát triển đầy động lực và đổi mới từ thập kỷ 1950 đến ngày nay. Năm 1958, Physicist William Higinbotham tạo ra trò chơi đầu tiên, Tennis for Two, một trong những bước khởi đầu cho sự phát triển của ngành công nghiệp game.

Thập kỷ 1970 chứng kiến sự phát triển mạnh mẽ của ngành game. Pong của Atari xuất hiện và trở thành một hiện tượng arcade, tạo nền tảng cho sự phát triển của các game arcade, thu hút được rất nhiều người chơi tại các trung tâm thương mại và khu vui chơi giải trí. Các trò chơi như Pac-Man (1980) và Donkey Kong (1981), đều là những đỉnh cao của kỷ nguyên arcade và trở thành biểu tượng của ngành game

Trong những năm 1980 và 1990, ngành làm game chuyển sang sử dụng các đĩa CD để lưu trữ dữ liệu và tăng cường đồ họa và âm thanh của các trò chơi. Những hệ máy chơi game nổi tiếng trong thời kỳ này bao gồm Nintendo Entertainment System, Sega Genesis và Sony PlayStation. Những trò chơi nổi bật trong thời kỳ này bao gồm Super Mario Bros., The Legend of Zelda, Sonic the Hedgehog và Final Fantasy I.

Trong thập kỷ 2000, sự bùng nổ của internet và công nghệ 3D đã tạo ra sự đa dạng trong thiết kế game. Các tựa game đầu tiên như Counter-Strike và Warcraft III đã mở ra kỷ nguyên của game đa người chơi trực tuyến và eSports.

Từ năm 2010 trở đi, game di động trở thành một lực lượng mạnh mẽ, với sự xuất hiện của các tựa game như Angry Birds, Flappy Bird, và Clash of Clans. Điều này mở ra một thế giới mới của giải trí trên thiết bị di động và tạo ra cơ hội mới cho các nhà phát triển game indie. Ngành công nghiệp cũng chứng kiến sự ra đời của các dịch vụ đám mây như Steam và Epic Games Store, đưa game trực tuyến và mô hình kinh doanh số lên một tầm cao mới. Game độc lập (indie) trở nên ngày càng phổ biến, với nhiều tựa game nhỏ có ảnh hưởng lớn đến cộng đồng game thủ.

Thời kỳ này còn đánh dấu sự nở rộ của thực tế ảo (VR) và thực tế tăng cường (AR) trong ngành công nghiệp game, với sự xuất hiện của các thiết bị như Oculus Rift và PlayStation VR. eSports, hay thể thao điện tử, cũng trở thành một hiện tượng toàn cầu, với

các sự kiện lớn và đội tuyển chuyên nghiệp thu hút sự chú ý của hàng triệu người hâm mộ trên thế giới.

1.2. Các công cụ làm game

Có nhiều công cụ và môi trường được sử dụng để tạo ra các trò chơi. Dưới đây là một số công cụ phổ biến mà nhà phát triển game thường sử dụng:

1. **Unity:** Là một trong những nền tảng phát triển game lớn nhất và phổ biến nhất. Unity hỗ trợ nhiều nền tảng, từ điện thoại di động đến máy tính và cả các thiết bị thực tế ảo.
2. **Unreal Engine:** Được phát triển bởi Epic Games, Unreal Engine là một công cụ mạnh mẽ cho việc tạo ra trò chơi với đồ họa 3D đỉnh cao. Nó cũng có hệ thống blueprint giúp người không chuyên có thể tạo game một cách dễ dàng.
3. **Godot Engine:** Một công cụ mã nguồn mở và miễn phí cho việc phát triển game 2D và 3D. Godot có sự đa dạng với nhiều tính năng như hệ thống vật lý, hệ thống ánh sáng, và nó hỗ trợ nhiều ngôn ngữ lập trình.
4. **Cocos2d-x:** Dành cho việc phát triển game di động, đặc biệt là game 2D. Nó hỗ trợ nhiều ngôn ngữ lập trình như C++, JavaScript, và Lua.
5. **GameMaker Studio:** Dành cho cả những người mới bắt đầu và những nhà phát triển kinh nghiệm. GameMaker chủ yếu được sử dụng để phát triển game 2D và có hệ thống Drag-and-Drop giúp việc phát triển dễ dàng hơn.
6. **Phaser:** Một framework JavaScript dành cho việc phát triển game trình duyệt (browser games) 2D. Nó giúp nhanh chóng xây dựng các trò chơi sử dụng HTML5 và JavaScript.
7. **RPG Maker:** Dành cho việc tạo ra các trò chơi nhập vai (RPG) một cách nhanh chóng và dễ dàng, đặc biệt là cho người mới bắt đầu.

Những công cụ này không chỉ giúp nhà phát triển tạo ra game mà còn hỗ trợ quản lý tài nguyên, lập trình, và kiểm thử game một cách hiệu quả.

1.3. Các ngôn ngữ làm game

Có nhiều ngôn ngữ lập trình được sử dụng để phát triển trò chơi, và sự chọn lựa phụ thuộc vào yêu cầu cụ thể của dự án cũng như sở thích và kinh nghiệm của nhà phát triển.

Dưới đây là một số ngôn ngữ phổ biến trong lĩnh vực làm game:

1. **C++:** Ngôn ngữ này thường được sử dụng cho việc phát triển game có yêu cầu về hiệu suất cao như game console và game PC. Unreal Engine, một trong những engine nổi tiếng, sử dụng C++ làm ngôn ngữ chính.
2. **C#:** Được sử dụng phổ biến trong phát triển game với Unity Engine. C# là ngôn ngữ lập trình chính cho Unity, và nó được ưa chuộng bởi sự dễ học và sử dụng.
3. **JavaScript/TypeScript:** Thường được sử dụng cho phát triển game web và game di động. Có nhiều framework như PhaserJS sử dụng JavaScript, còn TypeScript được sử dụng để phát triển với engine như Cocos2d-x.
4. **Python:** Mặc dù không phải là lựa chọn chính cho game đòi hỏi hiệu suất cao, Python vẫn được sử dụng cho việc phát triển game đơn giản và prototyping. Pygame là một thư viện phổ biến sử dụng Python cho việc làm game 2D.
5. **Java:** Được sử dụng chủ yếu trong việc phát triển game Android. LibGDX là một framework sử dụng Java cho việc làm game di động và desktop.
6. **Lua:** Thường được sử dụng làm ngôn ngữ kịch bản trong các game. Lua linh hoạt và dễ tích hợp, thường được sử dụng trong các engine như Unity, Unreal Engine, hoặc làm scripting language cho game.
7. **Haskell và Clojure:** Mặc dù ít phổ biến hơn, nhưng đã có dự án sử dụng Haskell và Clojure để phát triển game, chủ yếu làm các dự án nghiên cứu và thử nghiệm.

CHƯƠNG 2: TỔNG QUAN VỀ CÔNG CỤ LÀM GAME UNITY

2.1 Unity là gì?

Unity là một nền tảng phát triển game đa nền tảng (cross-platform) được phát triển bởi Unity Technologies. Được ra mắt lần đầu vào năm 2005, Unity đã nhanh chóng trở thành một trong những công cụ phát triển game phổ biến nhất trong ngành công nghiệp game.

Unity cho phép nhà phát triển tạo ra các trò chơi cho nhiều nền tảng khác nhau, bao gồm điện thoại di động, máy tính, console, cũng như các thiết bị thực tế ảo và thực tế tăng cường. Điểm độc đáo của Unity là khả năng phát triển game không chỉ cho môi trường Windows và macOS mà còn cho Linux.

Unity3D cung cấp một hệ thống toàn diện cho các lập trình viên, từ soạn thảo mã nguồn, xây dựng công cụ tự động hóa đến trình sửa lỗi nên cũng khá dễ sử dụng. Ngôn ngữ lập trình chính của Unity là C#, ngoài ra còn có hỗ trợ cho Javascript.

Unity cũng tận dụng chức năng của các thư viện phần mềm như engine mô phỏng vật lý PhysicX của Nvidia, OpenGL và Direct3D để kết xuất hình ảnh 3D, OpenAL cho âm thanh, ... nên nó hỗ trợ rất mạnh cho công việc lập trình game.



Hình 2.1: Logo Unity

2.2 Tính năng nổi bật của Unity

Trên thị trường game quốc tế và cả tại Việt Nam, Unity vẫn giữ vững vị thế của mình như một trong những game engine được nhiều nhà phát triển ưa chuộng. Điều này có lẽ đến từ những ưu điểm mà Unity mang lại. Trong đó có ba điểm mạnh quan trọng của Unity mà được nhiều nhà phát triển đánh giá cao:

1. Editor Thuận Tiện: Editor của Unity là một công cụ mạnh mẽ, giúp nhà phát triển không cần phải viết mã nguồn để sắp đặt các đối tượng trong game. Điều này giúp tối ưu

hóa quá trình phát triển, với khả năng kéo và thả, chỉnh sửa vị trí của các đối tượng trực tiếp trên Editor. Điều này không chỉ giúp tiết kiệm thời gian mà còn tạo ra sự linh hoạt và dễ dàng trong quá trình tạo nội dung.

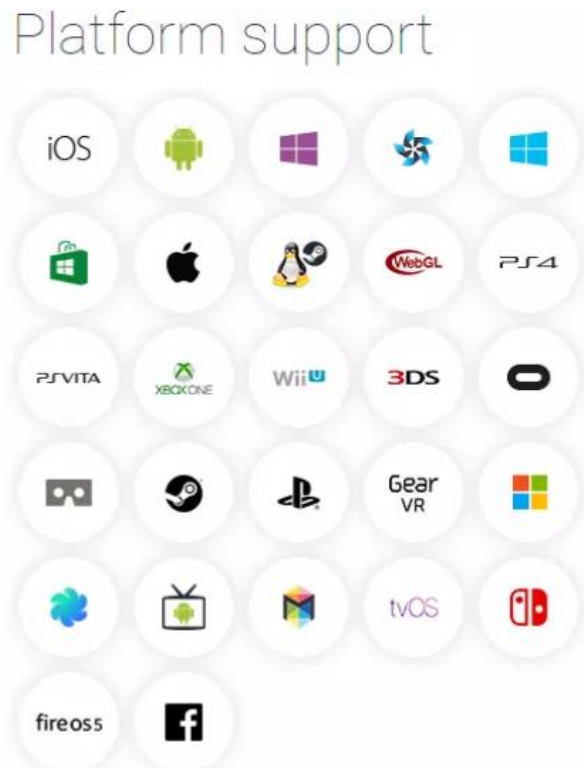
2. Đa Nền Tảng: Unity hỗ trợ gần như toàn bộ các nền tảng hiện có bao gồm: PlayStation 3, Xbox 360, Wii U, iOS, Android, Windows, Blackberry 10, OS X, Linux, trình duyệt Web và cả Flash. Khả năng đa nền tảng của Unity là một trong những đặc điểm quan trọng giúp nó nổi bật trong cộng đồng phát triển game. Việc tạo ra một trò chơi có thể chạy trên nhiều hệ điều hành và thiết bị khác nhau mang lại nhiều lợi ích quan trọng:

Nhà phát triển không cần phải xây dựng từ đầu cho mỗi nền tảng riêng biệt. Việc này giảm đáng kể thời gian và chi phí phát triển, vì một codebase có thể được tái sử dụng và chỉ cần điều chỉnh nhỏ để phù hợp với mỗi hệ điều hành hoặc thiết bị cụ thể.

Việc có thể đưa trò chơi của bạn đến nhiều người chơi trên nhiều nền tảng khác nhau tăng cơ hội tiếp cận đối tượng khách hàng. Điều này đặc biệt quan trọng trong thế giới ngày nay, khi người chơi sử dụng nhiều thiết bị khác nhau để trải nghiệm game.

Với khả năng chơi trên nhiều nền tảng, Unity là một công cụ linh hoạt cho cả những nhà phát triển đang làm việc trên PC, Mac, hay thậm chí là Linux. Điều này tạo điều kiện thuận lợi để xây dựng cộng đồng đa dạng và đa quốc gia.

Công ty và nhà phát triển có thể tận dụng sức mạnh của Unity để phát triển nhiều dự án khác nhau mà không cần phải mất công làm lại từ đầu. Điều này tăng tính hiệu quả và sự linh hoạt trong việc quản lý và phát triển nhiều sản phẩm game cùng một lúc.



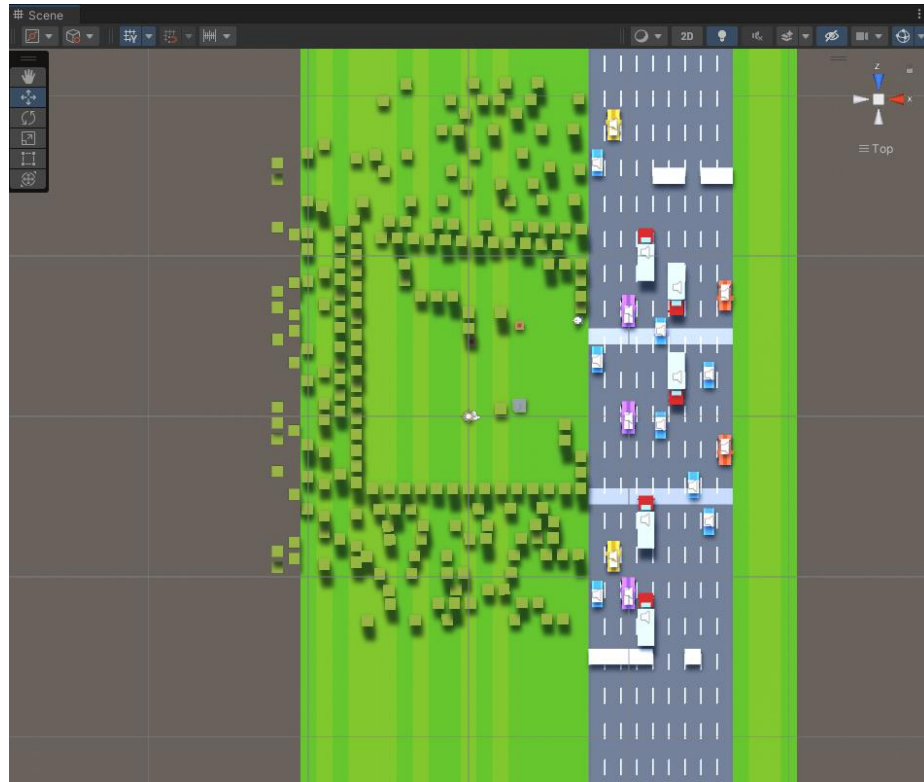
Hình 2.2: Các nền tảng hỗ trợ Unity

3. Tính kinh tế cao: Trong khi mang lại những tính năng mạnh mẽ, Unity vẫn giữ chính sách miễn phí đối với một số tính năng cơ bản. Điều này thu hút nhiều nhà phát triển, đặc biệt là những người mới bắt đầu hoặc các dự án nhỏ. Tuy nhiên, với các dự án miễn phí, việc xuất hiện logo Unity trong game là điều bắt buộc, tạo ra một hình ảnh thương hiệu đồng thời với việc cung cấp lợi ích miễn phí. Đến 1/1/2024, Unity thực hiện chính sách thu phí mới, tuy nhiên Unity vẫn giữ một sự linh hoạt nhất định để hỗ trợ cộng đồng game phát triển, đặc biệt là các doanh nghiệp vừa và nhỏ, trong khi vẫn thu được một khoản phí hợp lý từ các dự án phát triển thành công.

2.3. Các thành phần cơ bản trong Unity Editor

2.3.1 Scene View

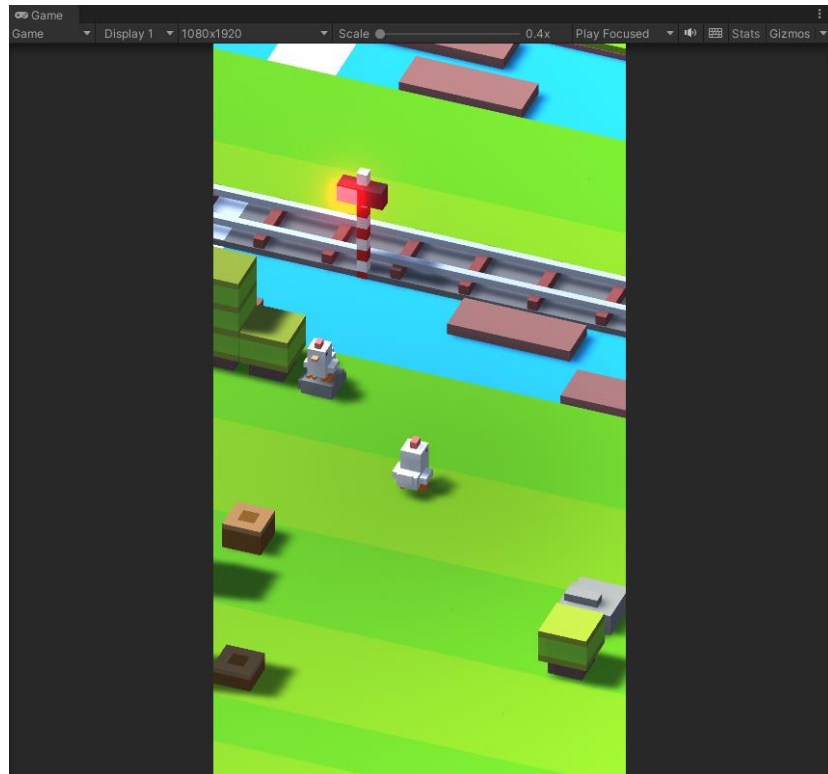
Phần này phân hiển thị các đối tượng trong scenes một cách trực quan, cho phép nhà phát triển xem và chỉnh sửa các đối tượng trong không gian 3D hoặc 2D. Scene View là nơi quan trọng để sắp đặt và kiểm tra vị trí của các đối tượng trong trò chơi.



Hình 2.3: Scene View

2.3.2 Game View

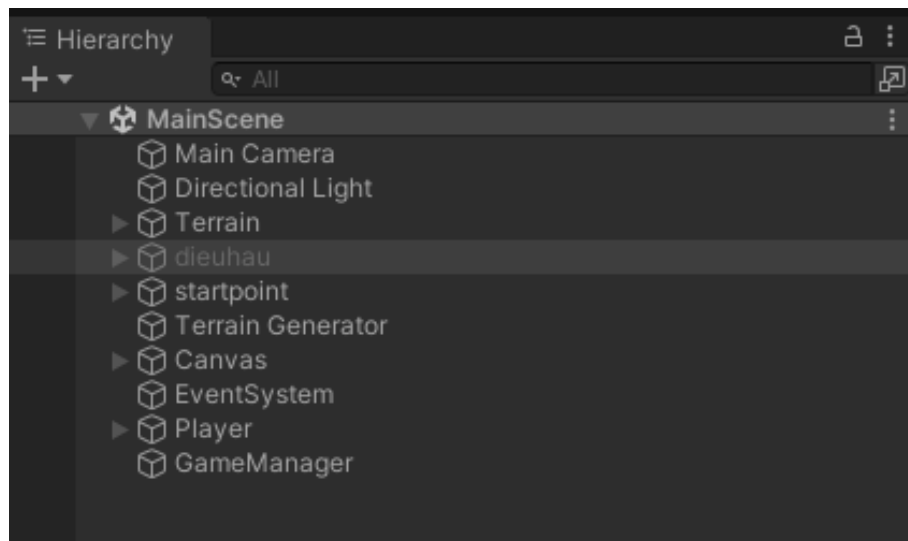
Hiện thị trò chơi trong quá trình chạy để nhà phát triển có cái nhìn thực tế về trải nghiệm người chơi. Game View giúp kiểm tra hiệu suất và thử nghiệm trò chơi trong môi trường chạy thực tế



Hình 2.4: Game View

2.3.3 Hierarchy Window

Liệt kê tất cả các đối tượng trong Scene và hiển thị cấu trúc hệ thống của chúng. Hierarchy Window cho phép nhà phát triển quản lý và tương tác với các đối tượng trong trò chơi.



Hình 2.5: Hierarchy Window

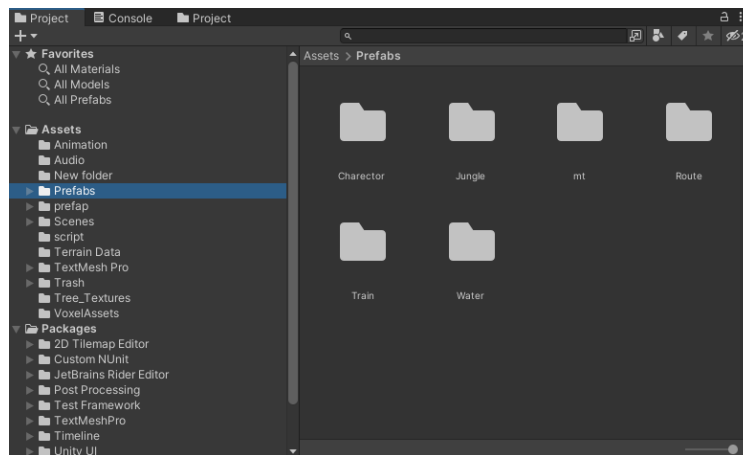
2.3.4 Project Window

Hiện thị tất cả các tài nguyên trong dự án như hình ảnh, âm thanh, mã nguồn, và prefab. Project Window giúp nhà phát triển quản lý và tổ chức tài nguyên của họ, cung cấp khả năng tìm kiếm, sắp xếp, và xem trước các file.

Cho phép nhà phát triển thực hiện các thao tác kéo thả giữa Project Window và các thành phần khác như Scene View hay Inspector Window

Cho phép nhà phát triển có thể tạo mới prefab, material, script, và nhiều loại asset khác

Project Window giúp tạo ra môi trường làm việc có tổ chức và thuận tiện, giúp nhà phát triển dễ dàng truy cập và thực hiện các thao tác liên quan đến tài nguyên của dự án Unity.

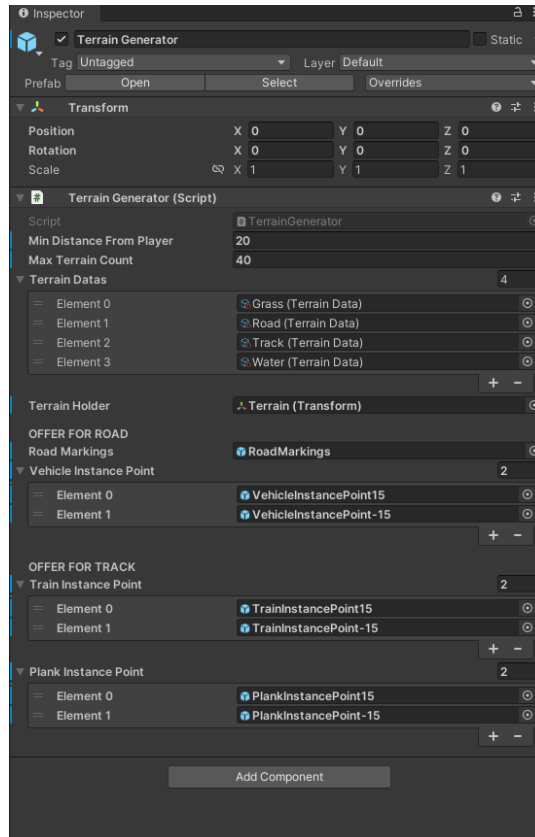


Hình 2.6: Project Window

2.3.5. Inspector Window

Hiện thị thông tin chi tiết về thuộc tính và những component của đối tượng được chọn trong scene hoặc project. Inspector Window là nơi nhà phát triển có thể điều chỉnh các thuộc tính của đối tượng, điều chỉnh, thêm và xóa các component. Ngoài ra Inspector Window cũng cung cấp khả năng xem trước và chỉnh sửa prefab trực tiếp từ scene.

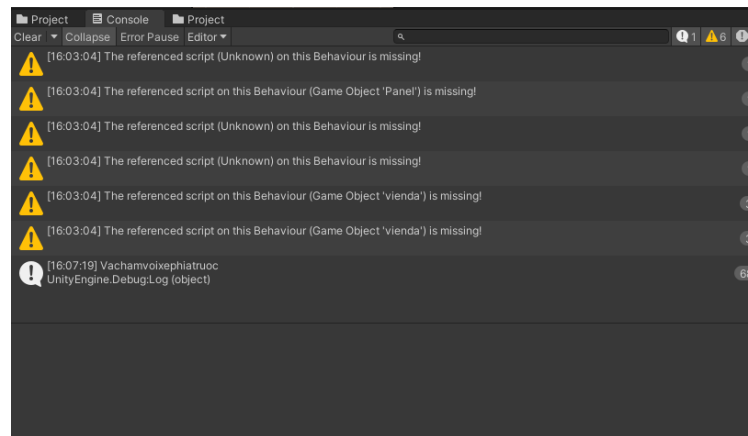
Đây là công cụ quan trọng giúp kiểm soát và tinh chỉnh đối tượng một cách hiệu quả trong quá trình phát triển trò chơi.



Hình 2.7: Inspector Window

2.3.6. Console Window

Hiển thị thông báo, cảnh báo, và lỗi từ trò chơi. Console Window là nơi quan trọng để gỡ lỗi và theo dõi hoạt động của trò chơi trong quá trình phát triển.



Hình 2.8 : Console Window

2.4. Một số khái niệm cơ bản trong unity

2.4.1. *GameObject*

GameObject trong Unity là đại diện cơ bản cho mọi đối tượng trong trò chơi từ nhân vật, đến ánh sáng và camera, có thể chứa các thuộc tính và các Component như Collider hay Script để điều khiển hành vi. Tất cả GameObject trong dự án được quản lý trong Hierarchy Window.

2.4.2. *Prefabs*

Prefabs trong Unity là các đối tượng hoặc cấu trúc được tạo sẵn cho phép nhà phát triển tái sử dụng chúng trong nhiều phần của trò chơi, giúp tối ưu hóa quá trình phát triển và duy trì tính đồng nhất. Chúng cho phép chỉnh sửa toàn bộ, tự động áp dụng các thay đổi cho tất cả các đối tượng Prefab tương tự, và được quản lý và hiển thị trong Project Window để dễ dàng theo dõi và tổ chức. Prefabs là công cụ quan trọng giúp quản lý và tái sử dụng các thành phần cấu trúc trong môi trường phát triển game của Unity.

2.4.3. *Assets*

Trong Unity, "Assets" là thuật ngữ được sử dụng để mô tả tất cả các tài nguyên được sử dụng trong một dự án game. Assets bao gồm mọi thứ từ hình ảnh, âm thanh, video, đến 3D models, script, và các file dữ liệu cần thiết để phát triển một trò chơi.

Tất cả các Assets được quản lý và hiển thị trong Project Window của Unity, nơi nhà phát triển có thể tổ chức, tìm kiếm, và thao tác các tài nguyên của họ

2.4.4. *Sprite*

Sprite trong Unity là một hình ảnh hoặc một đối tượng hình vẽ được sử dụng để hiển thị và tạo đối tượng 2D trong trò chơi. Chúng thường được sử dụng cho nhân vật hoặc các phần tử 2D khác trong môi trường game, đóng vai trò quan trọng trong việc tạo ra đồ họa cho game 2D.

2.4.5. *Shader và Material*

Giữa materials và shaders có mối liên kết chặt chẽ với nhau với nhau, điều đó giúp Unity tạo ra mô hình linh hoạt và mạnh mẽ cho việc xử lý đồ họa trong trò chơi

Shader trong Unity là 1 mã nguồn viết bằng ngôn ngữ HLSL, được thiết kế để kiểm soát cách mà GPU xử lý ánh sáng, màu sắc và hiệu ứng trên đối tượng trong trò chơi.

Material sử dụng một Shader cụ thể để xác định cách nó tương tác với ánh sáng và màu sắc. Khi gán một Material lên một đối tượng, đối tượng đó sẽ áp dụng logic xử lý của Shader liên kết với Material.

2.4.6. Camera

Trong Unity, Camera là một thành phần (component) quan trọng được gán vào 1 gameobject để xác định tầm nhìn và hiển thị môi trường game

2.4.7. Component

Component trong Unity là các module tái sử dụng. Unity cung cấp nhiều loại Component như Collider (xử lý va chạm va chạm), Renderer (để xác định cách đối tượng được hiển thị), Script (mã nguồn điều khiển),.. được gán liền với GameObject để mở rộng chức năng và hành vi của chúng. Chúng giúp tăng tính tái sử dụng, quản lý và chia sẻ giữa các đối tượng, và chứa logic cụ thể, như xử lý sự kiện và điều khiển hành vi, làm nền tảng cho tạo ra các đối tượng động và tương tác trong môi trường game.

2.4.8. Script

Trong Unity, "Script" là một đoạn mã nguồn được viết bằng ngôn ngữ lập trình như C#, JavaScript hoặc Boo, có nhiệm vụ điều khiển hành vi và tương tác của đối tượng trong trò chơi.

2.4.9. Transform

Transform là Component quan trọng xác định vị trí, quay, và tỷ lệ của một GameObject trong không gian 3D.

2.4.10. Collider

Collider là một Component để xác định khu vực vật lý của một GameObject, giúp kiểm tra khi nào nó tương tác với các đối tượng khác.

2.4.11. Rigidbody

Rigidbody là một thành phần trong Unity được sử dụng để thêm tính chất vật lý và chuyển động động lực vào GameObject. Nó giúp đối tượng phản ứng với lực như trọng lực và lực va chạm. Rigidbody cần được gán kèm với Collider để xử lý va chạm vật lý. Rigidbody cũng có thể được sử dụng để áp dụng lực và chuyển động thông qua mã nguồn Script. Việc sử dụng chính dữ liệu và tính năng của Rigidbody là quan trọng để mô phỏng chính xác vật lý trong trò chơi Unity.

2.5. Các hàm API phổ biến thường được sử dụng trong Unity

2.5.1. *MonoBehaviour*

MonoBehaviour là một lớp trong Unity, mô tả các đối tượng có thể được gắn kết vào các GameObject trong môi trường phát triển game của Unity. MonoBehaviour là cơ sở cho hầu hết các component trong Unity, bao gồm các phương thức để quản lý vòng đời của đối tượng.

MonoBehaviour là một phần quan trọng của cách Unity xử lý logic của game và cung cấp một cách thuận tiện để quản lý vòng đời của đối tượng trong game.

Dưới đây là một số phương thức quan trọng của MonoBehaviour:

Awake(): Phương thức này được gọi khi đối tượng được tạo ra. Nó thường được sử dụng để khởi tạo các biến hoặc cài đặt trạng thái ban đầu.

OnEnable(): Được gọi khi đối tượng được kích hoạt hoặc bật. Nó được gọi sau khi đối tượng được tạo ra và trước khi Start() được gọi. Thường được sử dụng để thực hiện các cài đặt hoặc thay đổi trạng thái của đối tượng khi nó được kích hoạt, hay thực hiện các công việc cần thiết khi đối tượng chuyển từ trạng thái không hoạt động sang trạng thái hoạt động.

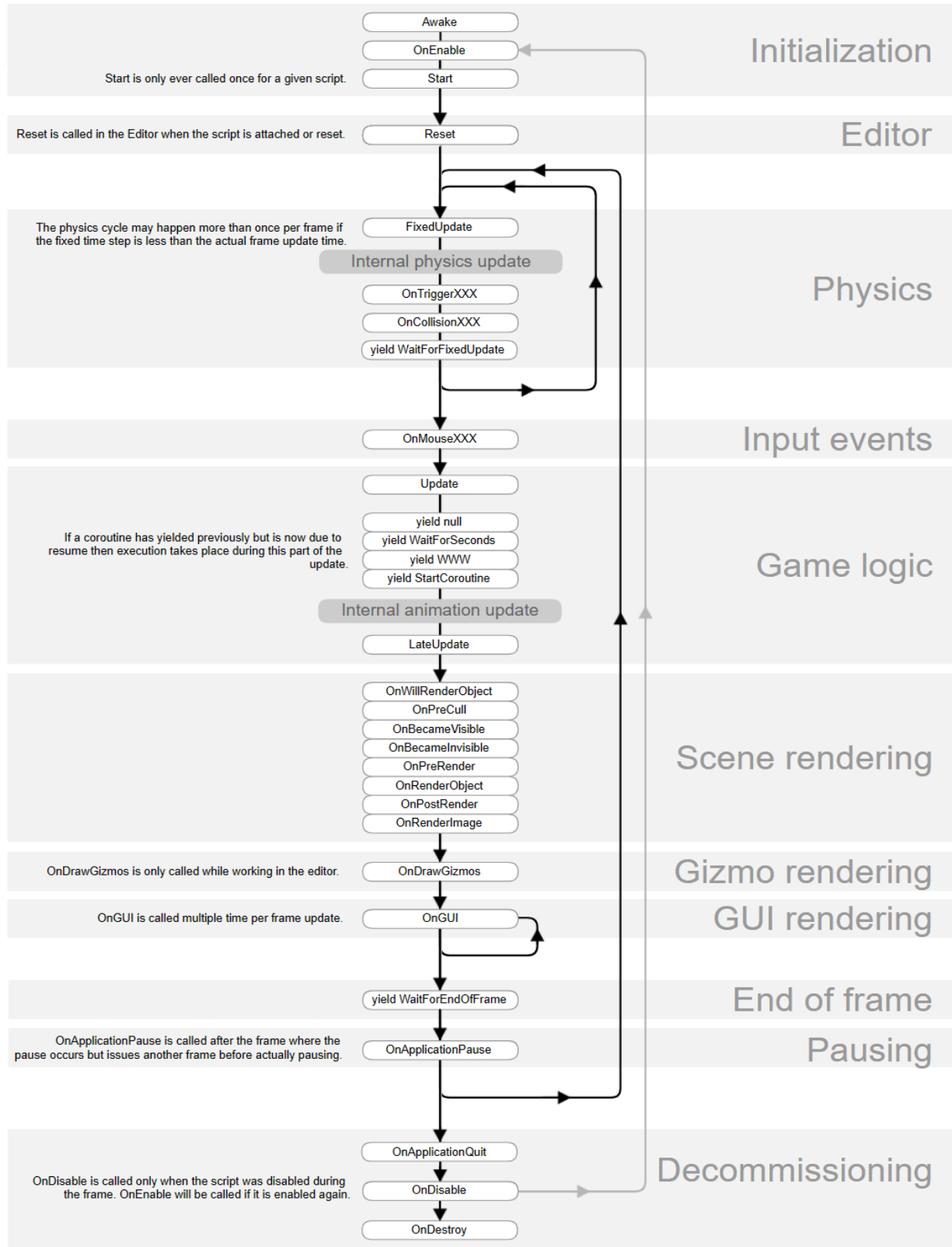
Start(): Được gọi trước khi cảnh bắt đầu chạy. Thường được sử dụng để chuẩn bị dữ liệu hoặc tìm kiếm các đối tượng khác trong cảnh.

Update(): Được gọi mỗi frame. Đây là nơi chính để đặt mã logic của game. Thời gian giữa các lần gọi Update() không đồng đều và phụ thuộc vào tốc độ khung hình của máy tính. Thường được sử dụng để đặt mã logic của game mà không liên quan đến vật lý.

FixedUpdate(): Tương tự như Update(), nhưng được gọi vào các khoảng thời gian cố định dựa trên vật lý (ví dụ: mỗi 0.02 giây). Thường được sử dụng để xử lý các tác vụ vật lý.

LateUpdate(): Được gọi sau khi tất cả các Update() đã được gọi. Thường được sử dụng khi cần đảm bảo rằng tất cả các thay đổi vị trí và trạng thái khác đã được cập nhật.

OnDestroy(): Được gọi khi đối tượng sẽ bị hủy bỏ. Thường được sử dụng để dọn dẹp tài nguyên và các thao tác khác trước khi đối tượng biến mất.



Hình 2.9: Thứ tự thực thi trong Unity

2.5.2. GameObject

Instantiate(): Tạo một thể hiện mới của một GameObject.

Destroy(): Hủy bỏ một GameObject hoặc Component.

SetActive(): Kích hoạt hoặc vô hiệu hóa một GameObject.

2.5.3. Transform

Translate(): Di chuyển vị trí của GameObject.

Rotate(): Quay GameObject theo một trục cụ thể.

Scale(): Thay đổi kích thước của GameObject.

2.5.4. Input

Input.GetKey(), Input.GetKeyDown(), Input.GetKeyUp(): Kiểm tra trạng thái của các phím.

Input.GetMouseButtonDown(), Input.GetMouseButtonUp(), Input.mousePosition: Xử lý sự kiện chuột.

2.5.5. Physics

Physics.Raycast(): Thực hiện raycasting để kiểm tra va chạm.

Rigidbody.AddForce(): Thêm lực cho một đối tượng có Rigidbody.

2.5.6. Colliders

OnCollisionEnter(), OnCollisionStay(), OnCollisionExit(): Xử lý va chạm giữa các Collision. Khi một Collision xảy ra, Unity sẽ cung cấp thông tin về va chạm, như vị trí va chạm, hướng của va chạm và thông tin về các đối tượng tham gia.

OnTriggerEnter(), OnTriggerStay(), OnTriggerExit(): Xử lý sự kiện trigger giữa các Trigger. Không có va chạm vật lý nào xảy ra khi sử dụng Trigger. Thay vào đó, thông tin về va chạm chỉ làm kích hoạt sự kiện

2.5.7. Audio

AudioSource.Play(), AudioSource.Stop(), AudioSource.volume: Quản lý âm thanh.

2.5.8. SceneManager

SceneManager.LoadScene(): Chuyển đổi giữa các cảnh trong game.

2.5.9. Time

Time.deltaTime: Thời gian giữa hai frame, thường được sử dụng để làm cho các chuyển động mượt mà và không phụ thuộc vào tốc độ khung hình.

2.5.10. Coroutine

StartCoroutine(), StopCoroutine(): Sử dụng để xử lý các công việc không đồng bộ trong một khoảng thời gian dài hơn mà không làm đơ game.

2.5.11. ScriptableObject

Scriptable Object trong Unity là một lớp được sử dụng để tạo và lưu trữ dữ liệu cố định mà không cần liên kết với GameObject nào trong cảnh. Dữ liệu này có thể được sử dụng để lưu trữ cấu hình, giá trị cố định, hoặc bất kỳ loại dữ liệu nào cần phải được chia sẻ và tái sử dụng giữa các script và các phần khác của game. Các đối tượng ScriptableObject có thể được tạo và quản lý trong trình đơn Asset của Unity và có thể được truy cập từ các script khác trong game để sử dụng thông tin chúng chứa. Việc sử dụng ScriptableObject giúp cải thiện quản lý dữ liệu và tổ chức mã nguồn của dự án.

CHƯƠNG 3: GAME CROSSY ROAD SỬ DỤNG UNITY

3.1 Phân tích thiết kế game

3.1.1. Đồ họa

Game sử dụng phong cách đồ họa pixel art, voxel với màu sắc đơn giản, tươi sáng, nổi bật và hấp dẫn. Điều này tạo ra một trải nghiệm trực quan độc đáo và làm tăng tính linh hoạt của trò chơi.

Game có nhiều môi trường khác nhau, từ con đường, sông, đến rừng. Mỗi môi trường đều có đặc điểm và thách thức riêng, và đồ họa được thiết kế để phản ánh sự đa dạng này.

3.1.2. Âm thanh

Âm thanh trong game tương thích với phong cách pixel art, voxel và tạo ra một trải nghiệm âm thanh thoải mái và vui nhộn. Các âm thanh bao gồm nhạc nền nhẹ nhàng và hài hước, cũng như các hiệu ứng âm thanh vui nhộn khi người chơi điều khiển nhân vật qua các chướng ngại vật, tạo ra môi trường âm thanh thú vị và làm tăng tính giải trí của trò chơi.

3.1.3. Giao diện

Giao diện người dùng của game được thiết kế đơn giản và thân thiện, với các nút và biểu tượng dễ hiểu. Điều này giúp người chơi dễ dàng thao tác và tập trung vào trò chơi mà không bị phân tâm bởi yếu tố phức tạp của giao diện.

3.1.4. Điều khiển

Game hỗ trợ hai loại điều khiển là bàn phím với bộ nút W,A,S,D(đối với máy tính) và chạm vuốt (đối với điện thoại)

3.1.5. Nhiệm vụ:

Nhiệm vụ trong game là điều khiển nhân vật của bạn để vượt qua đường và chướng ngại vật một cách an toàn. Trong quá trình di chuyển, người chơi cần tránh xe cộ, tàu hỏa, sông, và các chướng ngại vật khác. Mục tiêu là đi được càng xa càng tốt và đạt được điểm số cao. Nhiệm vụ cơ bản là thử thách sự nhạy bén và phản xạ của người chơi để vượt qua các tình huống khó khăn và đạt được điểm số cao nhất có thể

3.1.6. Map

Map trong game được thiết kế để tạo ra một môi trường đa dạng và thách thức cho người chơi. Map bao gồm các terrain như đường, đường rừng, sông, và đường sắt. Các

terrain sẽ có kích thước giống nhau với tỷ lệ scale theo x,y,x là 1,1,50. Mỗi terrain có những đặc điểm riêng, với các chướng ngại vật và tình huống nguy hiểm khác nhau, yêu cầu người chơi phải thích ứng và phản xạ nhanh chóng để tránh va chạm và giữ cho nhân vật sống sót:

Đường(Road):

Đặc điểm: Không gây nguy hiểm

Chướng ngại vật loại gây nguy hiểm: Các loại xe

Rừng(Jungle):

Đặc điểm: Không gây nguy hiểm

Chướng ngại vật loại không gây nguy hiểm: Cây, đá, khúc gỗ

Sông(River):

Đặc điểm: Gây nguy hiểm

Chướng ngại vật loại người chơi có thể đứng lên đó: planket, duckweed

Đường sắt(Track):

Đặc điểm: Không gây nguy hiểm

Chướng ngại vật loại cảnh báo: Đèn báo

Chướng ngại vật loại gây nguy hiểm: Tàu hỏa

Map được tạo ra từ các terrain theo cách ngẫu nhiên, và các chướng ngại vật trên mỗi terrain cũng được sinh ngẫu nhiên, điều này tạo ra sự độc đáo và không đoán trước được trong mỗi ván chơi, tạo ra một trải nghiệm đơn giản nhưng đầy thách thức, với sự đổi mới và sự ngẫu nhiên sẽ giữ cho người chơi liên tục quan tâm và thử thách.

3.1.6. Góc nhìn

Góc nhìn trong game là góc nhìn từ trên xuống (top-down view) và Camera sử dụng phép chiếu Orthographic giúp người chơi dễ dàng quan sát và đối mặt với các chướng ngại vật, xe cộ, sông và các yếu tố khác trong trò chơi.

3.1.7. Hệ thống tiền tệ (gold) và skin nhân vật

Trong quá trình tạo map, sẽ có tỷ lệ nhỏ sinh ra những đồng gold, người chơi có thể thu thập những đồng gold đó bằng cách va chạm

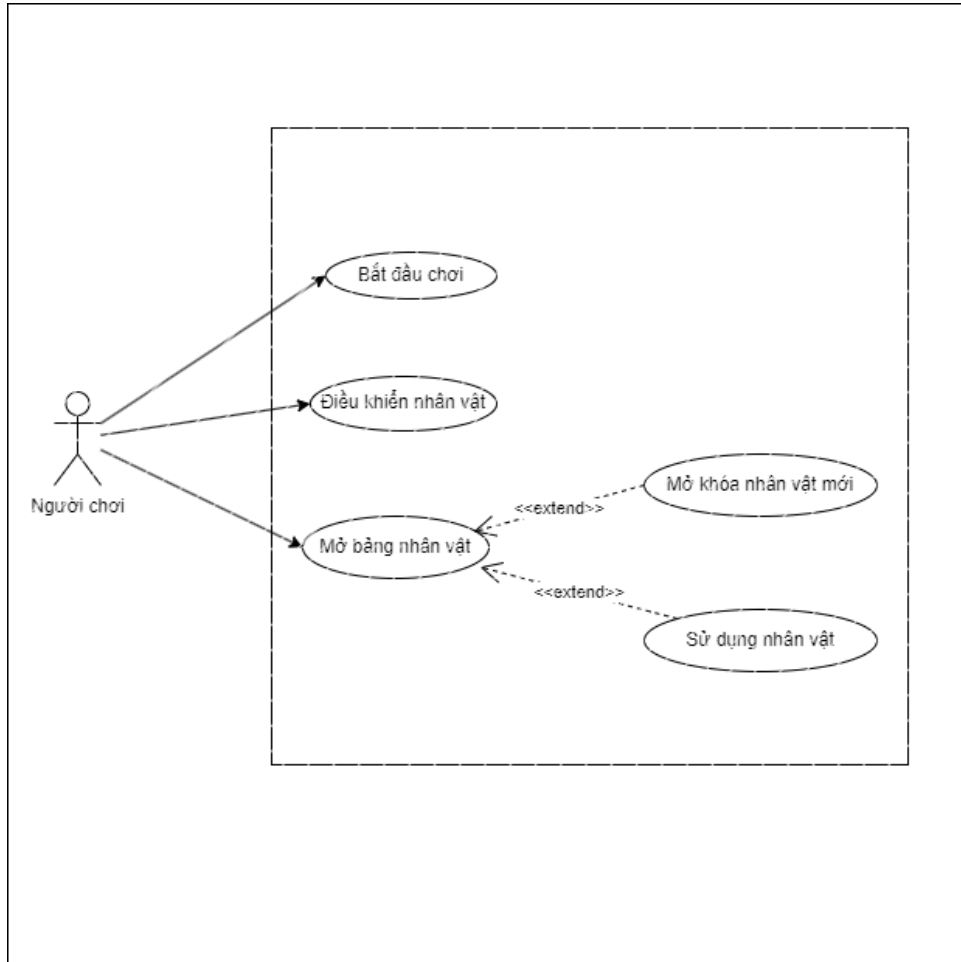
Người chơi có thể sử dụng những đồng gold tích lũy được để mở khóa các skin nhân vật đẹp mắt, thay đổi hình ảnh của nhân vật của mình trong trò chơi.

CHƯƠNG 4: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

4.1. Xác định tác nhân

Người chơi là tác nhân của game

4.2. Xây dựng biểu đồ Usecase



Hình 4. 1: Mô hình UseCase tổng quát

4.3. Danh sách use case:

- * Bắt đầu chơi
- * Điều khiển nhân vật
- * Mở bảng nhân vật
- * Mở khóa nhân vật mới
- * Sử dụng nhân vật

4.4. Đặc tả UseCase

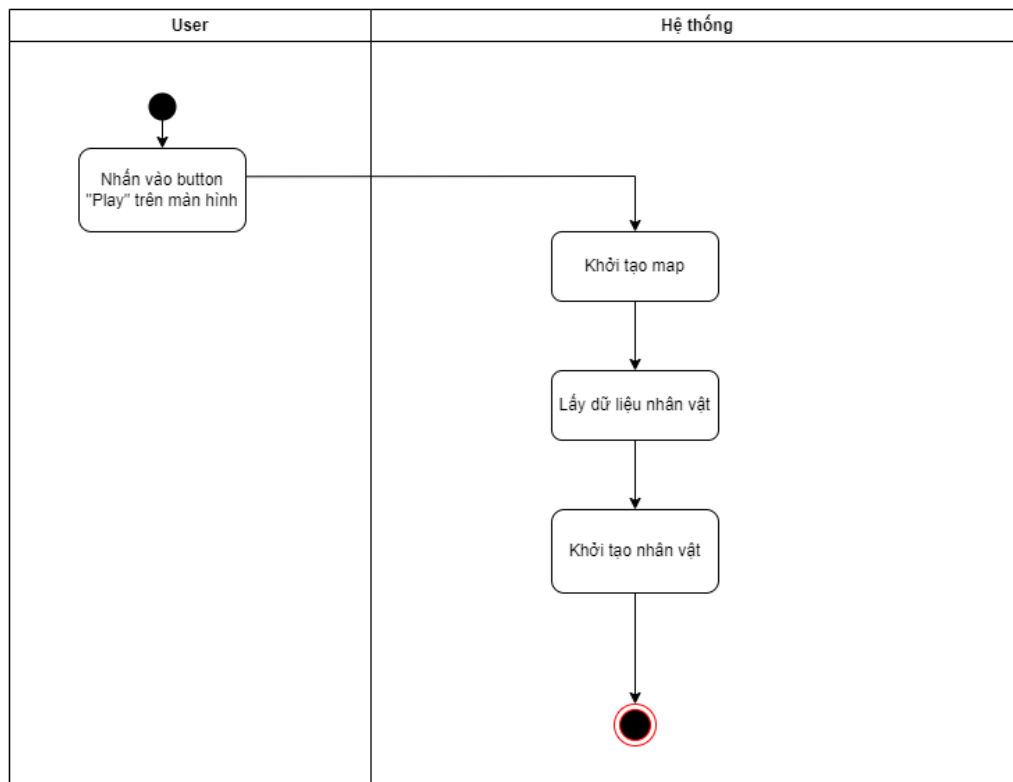
4.4.1 Use case bắt đầu chơi

* Kịch bản Use Case:

Bảng 4. 1: Kịch bản Use Case bắt đầu chơi

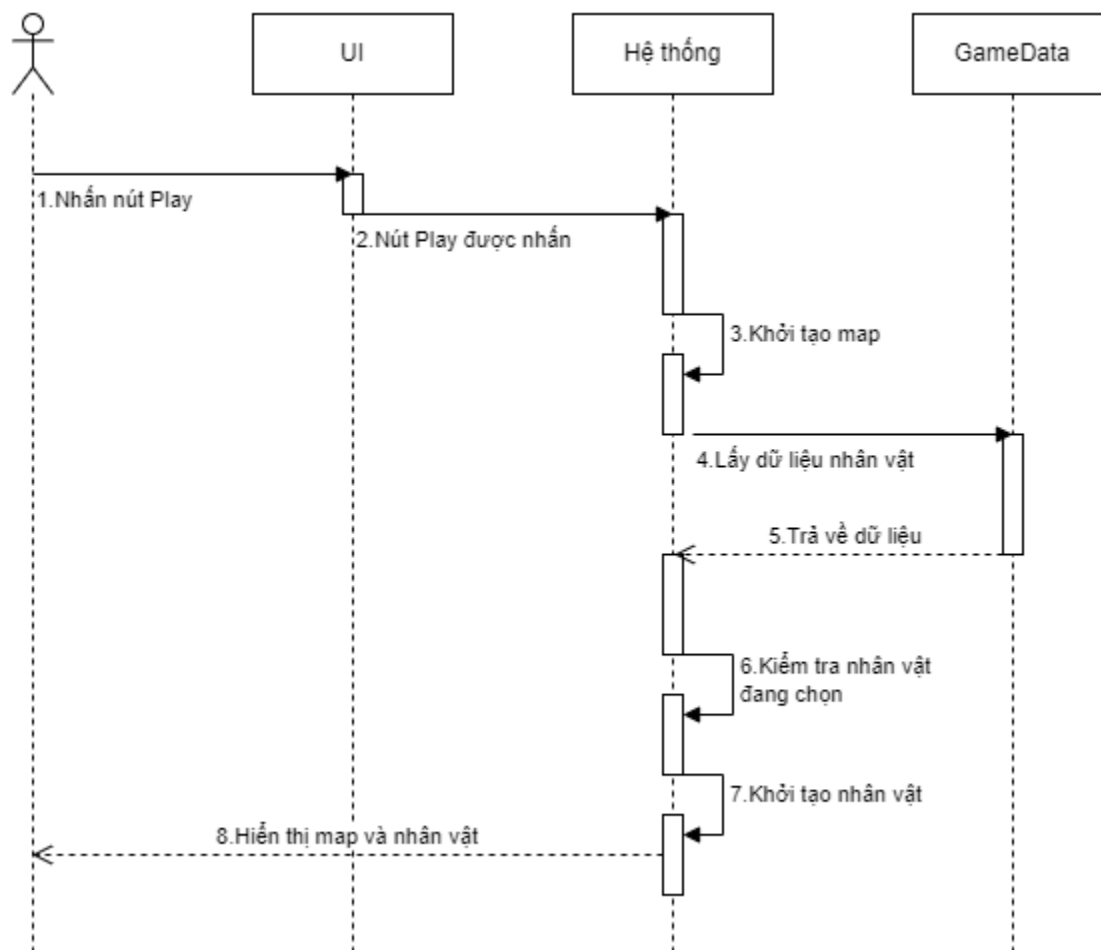
Mô tả	Người chơi bắt đầu chơi
Tác nhân	Người chơi
Tiền điều kiện	Có dữ liệu của map Có dữ liệu của nhân vật
Hậu điều kiện	Hiển thị map Hiển thị nhân vật
Luồng sự kiện	1. Người chơi nhấn vào button “Play” trên màn hình 3. Đóng cửa sổ Menu 4. Hệ thống tạo map 5. Hệ thống tạo nhân vật
Luồng ngoại lệ	

* Sơ đồ Activity:



Hình 4. 2: Sơ đồ Activity Use Case bắt đầu chơi

* Sơ đồ tuần tự



Hình 4. 3: Sơ đồ tuần tự Use Case bắt đầu chơi

4.4.2 Use case điều khiển nhân vật:

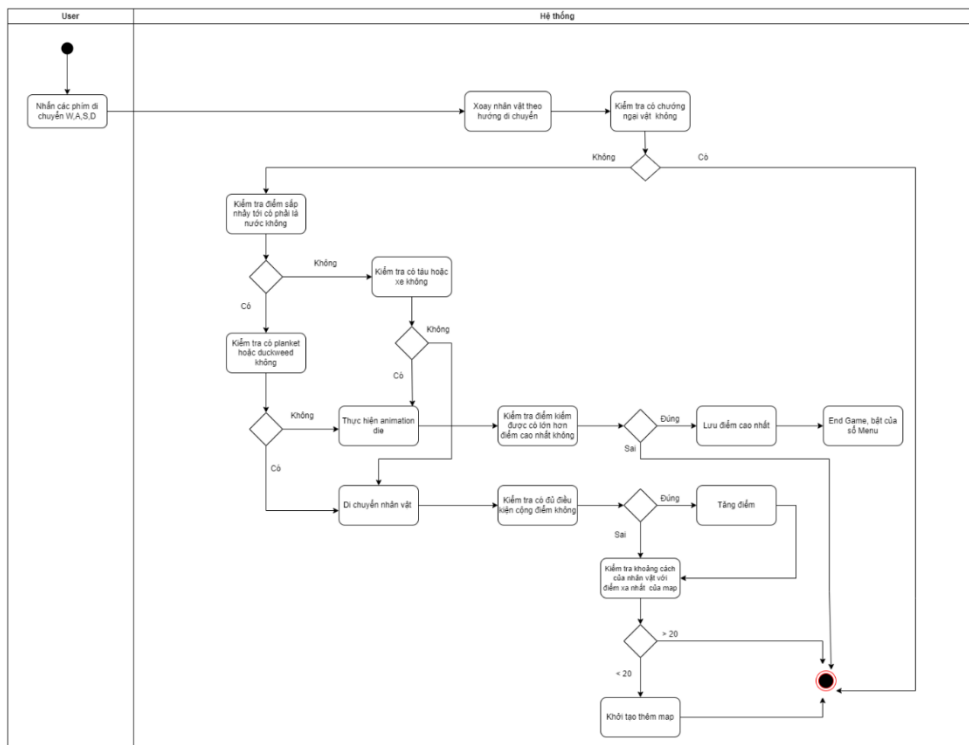
* Kịch bản Use Case:

Bảng 4. 2: Kịch bản Use Case điều khiển nhân vật

Mô tả	Người chơi điều khiển nhân vật
Tác nhân	Người chơi
Tiền điều kiện	Người chơi đã nhấn nút Play Người chơi nhấn các nút điều khiển W,A,S,D
Hậu điều kiện	Nhân vật di chuyển
Luồng sự kiện	1.Nhấn nút điều khiển 3.Nhân vật xoay về hướng di chuyển

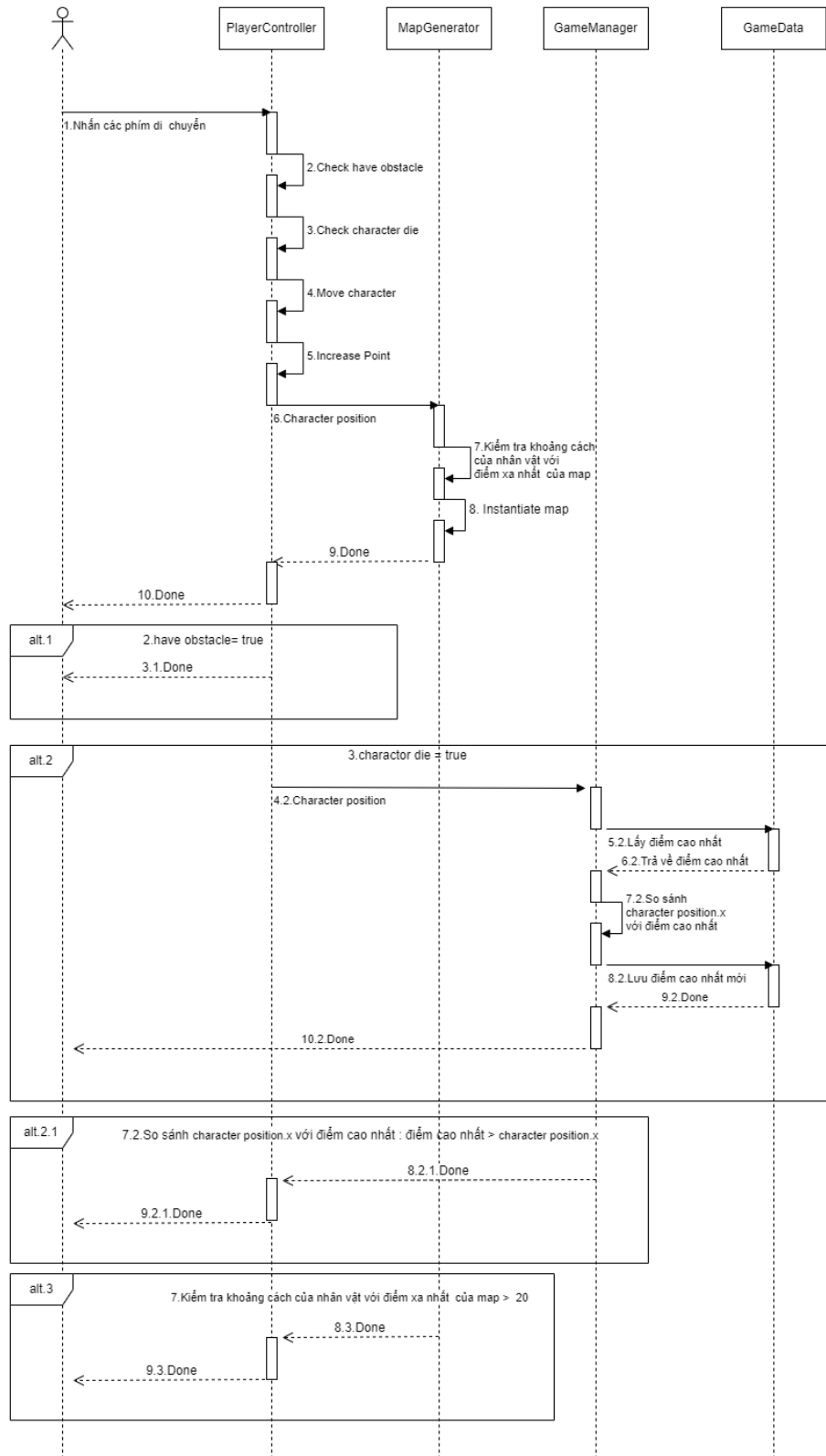
	3.Hệ thống kiểm tra không có chướng ngại vật theo hướng di chuyển không 4.Hệ thống kiểm tra vị trí di chuyển tới là điểm an toàn 5.Di chuyển nhân vật 6.Tăng điểm 7.Hệ thống kiểm tra map hiện tại so với vị trí của người chơi 8.Hệ thống khởi tạo thêm map
Luồng ngoại lệ	4a. Hệ thống kiểm tra vị trí di chuyển tới có là điểm nguy hiểm 5a.Thực hiện animation nhân vật die 6a.Hệ thống so sánh điểm kiếm được lớn hơn điểm cao nhất 7a.Hệ thống lưu điểm cao nhất 8a. Mở bật của sổ Menu

* Sơ đồ Activity:



Hình 4. 4: Sơ đồ Activity Use Case điều khiển nhân vật

* Sơ đồ tuần tự



Hình 4. 5: Sơ đồ tuần tự Use Case điều khiển nhân vật

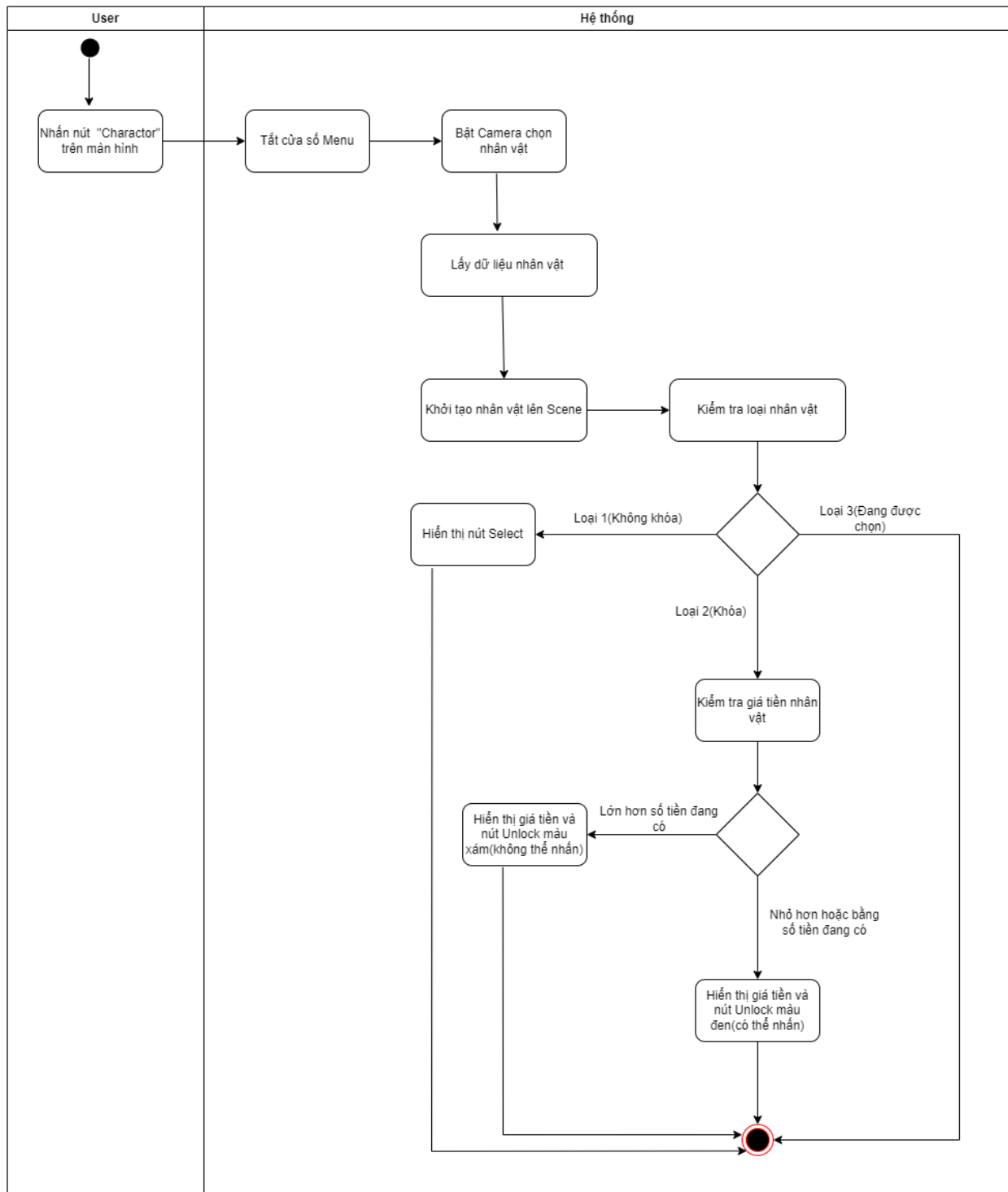
4.4.3 Use case mở bảng nhân vật:

* Kịch bản Use Case:

Bảng 4. 3: Kịch bản mở bảng nhân vật

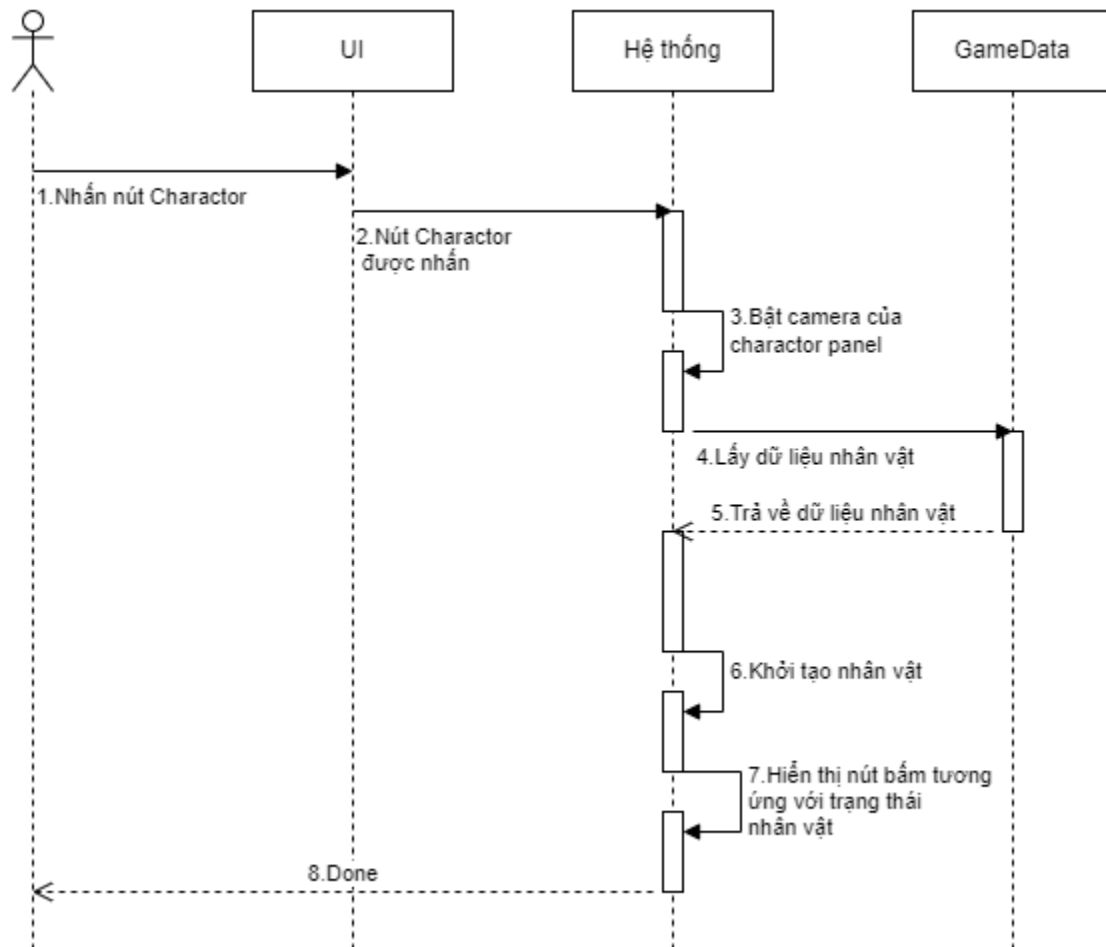
Mô tả	Người chơi mở bảng nhân vật
Tác nhân	Người chơi
Tiền điều kiện	Người chơi đã nhấn nút Charactor
Hậu điều kiện	Mở bảng nhân vật
Luồng sự kiện	1.Nhấn nút Character 2.Ấn cửa sổ Menu 3. Bật camera của bảng nhân vật 4.Lấy dữ liệu nhân vật 5.Khởi tạo prefab của nhân vật 6.Hiển thị nút bấm tương ứng với trạng thái của nhân vật
Luồng ngoại lệ	

* Sơ đồ Activity:



Hình 4. 6: Sơ đồ Activity Use Case mở bảng nhân vật

* Sơ đồ tuần tự:



Hình 4. 7: Sơ đồ tuần tự Use Case mở bảng nhân vật

4.4.4 Use case mở khóa nhân vật:

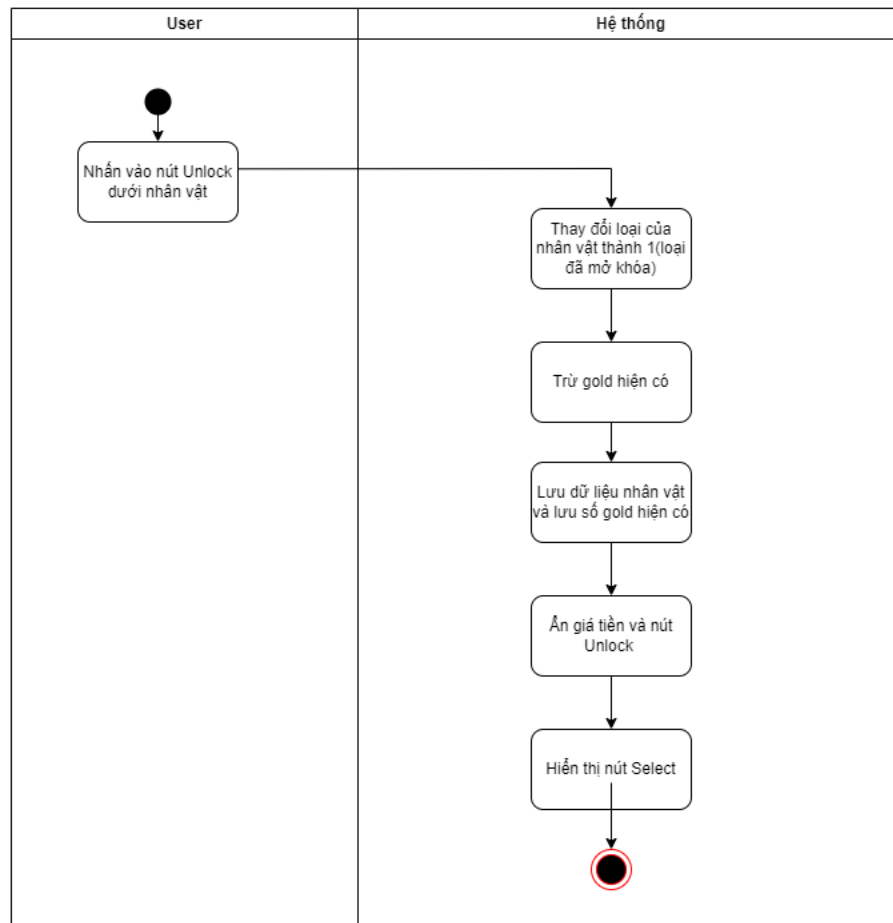
* Kịch bản Use Case:

Bảng 4. 4: Kịch bản mở bảng khóa nhân vật

Mô tả	Người chơi mở khóa nhân vật mới
Tác nhân	Người chơi
Tiền điều kiện	Người chơi đã mở bảng nhân vật

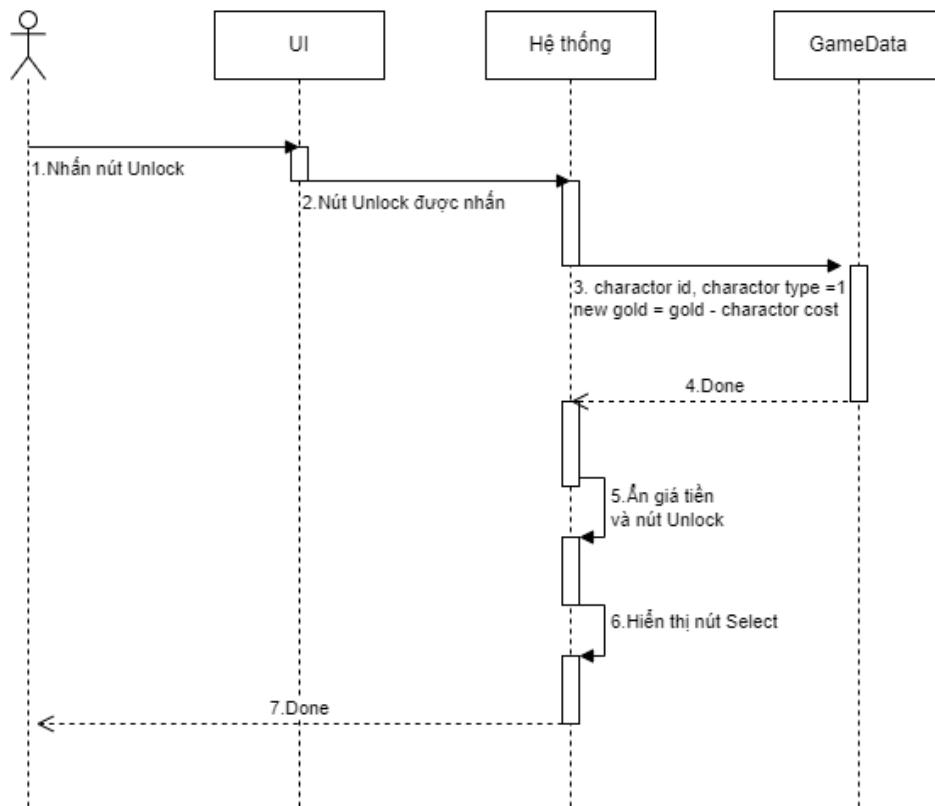
	Người chơi nhấn bấm vào nút Unlock dưới nhân vật
Hậu điều kiện	Mở khóa nhân vật
Luồng sự kiện	1. Nhấn nút Unlock 2. Hệ thống thay đổi trạng thái của nhân vật thành đã unlock 3. Hệ thống lưu dữ liệu nhân vật 4. Hệ thống trừ gold và lưu vào data 4. Ấn nút Unlock và hiển thị nút Select
Luồng ngoại lệ	

* Sơ đồ Activity:



Hình 4. 8: Sơ đồ Activity Use Case mở khóa nhân vật

* Sơ đồ tuần tự:



Hình 4. 9: Sơ đồ tuần tự Use Case mở khóa nhân vật

4.4.5 Use case chọn nhân vật:

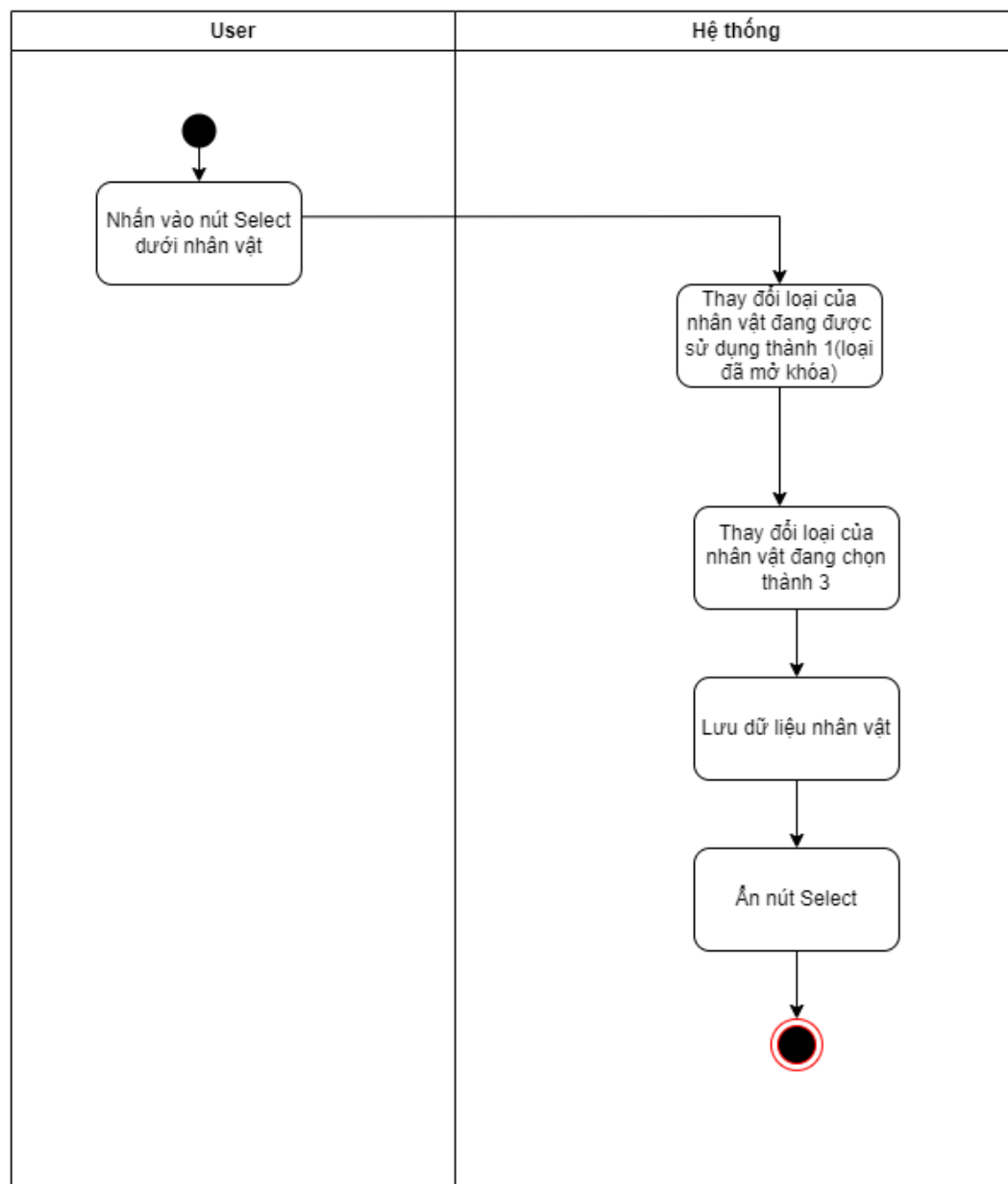
* Kịch bản Use Case:

Bảng 4. 5: Kịch bản chọn nhân vật

Mô tả	Người chơi chọn nhân vật
Tác nhân	Người chơi
Tiền điều kiện	Người chơi đã mở bảng nhân vật Người chơi nhấn bấm vào nút Select dưới nhân vật
Hậu điều kiện	Sử dụng nhân vật
Luồng sự kiện	1.Nhấn nút Select 2.Hệ thống thay đổi trạng thái của nhân vật đang được sử dụng thành đã unlock 3. Hệ thống thay đổi trạng thái của nhân vật đang chọn thành sử dụng 4.Hệ thống lưu dữ liệu nhân vật 4.Ấn nút Select

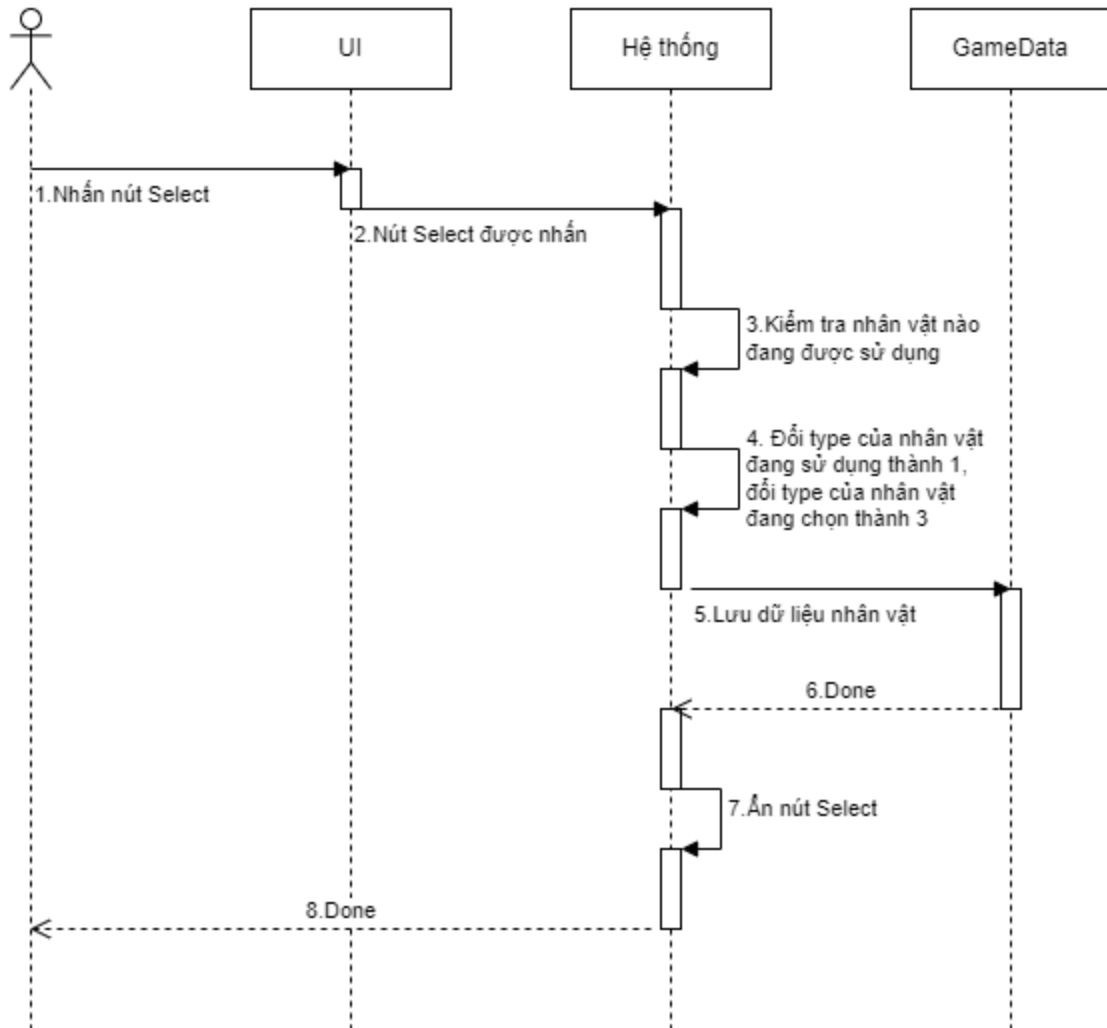
Luồng ngoại lệ	
-----------------------	--

* Sơ đồ Activity:



Hình 4. 10: Sơ đồ Activity Use Case chọn nhân vật

* Sơ đồ tuần tự:



Hình 4. 11: Sơ đồ tuần tự Use Case chọn nhân vật

4.5 Dữ liệu

Game sử dụng ScriptableObject để lưu dữ liệu

```

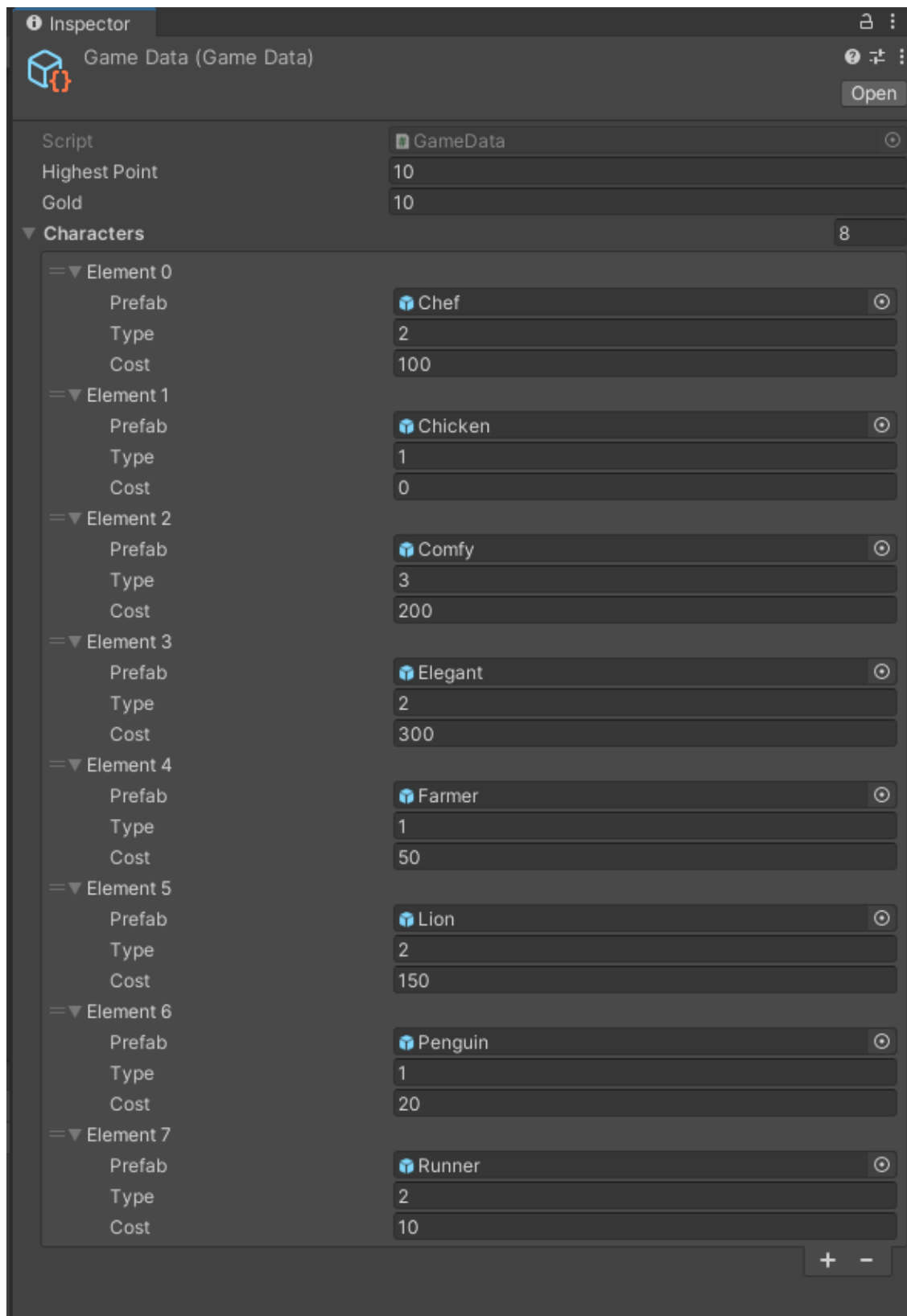
public class GameData : ScriptableObject
{
    public int highestPoint;
    public int gold;
    public List<Character> characters = new List<Character>();
}
  
```

Trong đó:

- * highestPoint: là tổng số điểm cao nhất mà người chơi kiếm được
- * gold: Là số gold hiện có
- * characters: là 1 danh sách các Character hiện có

```
public struct Character
{
    public GameObject prefab;
    public int status; // 1 lock , 2 unlock, 3 select
    public int cost;
    public Character(GameObject prefab, int type, int cost)
    {
        this.prefab = prefab;
        this.type = type;
        this.cost = cost;
    }
}
```

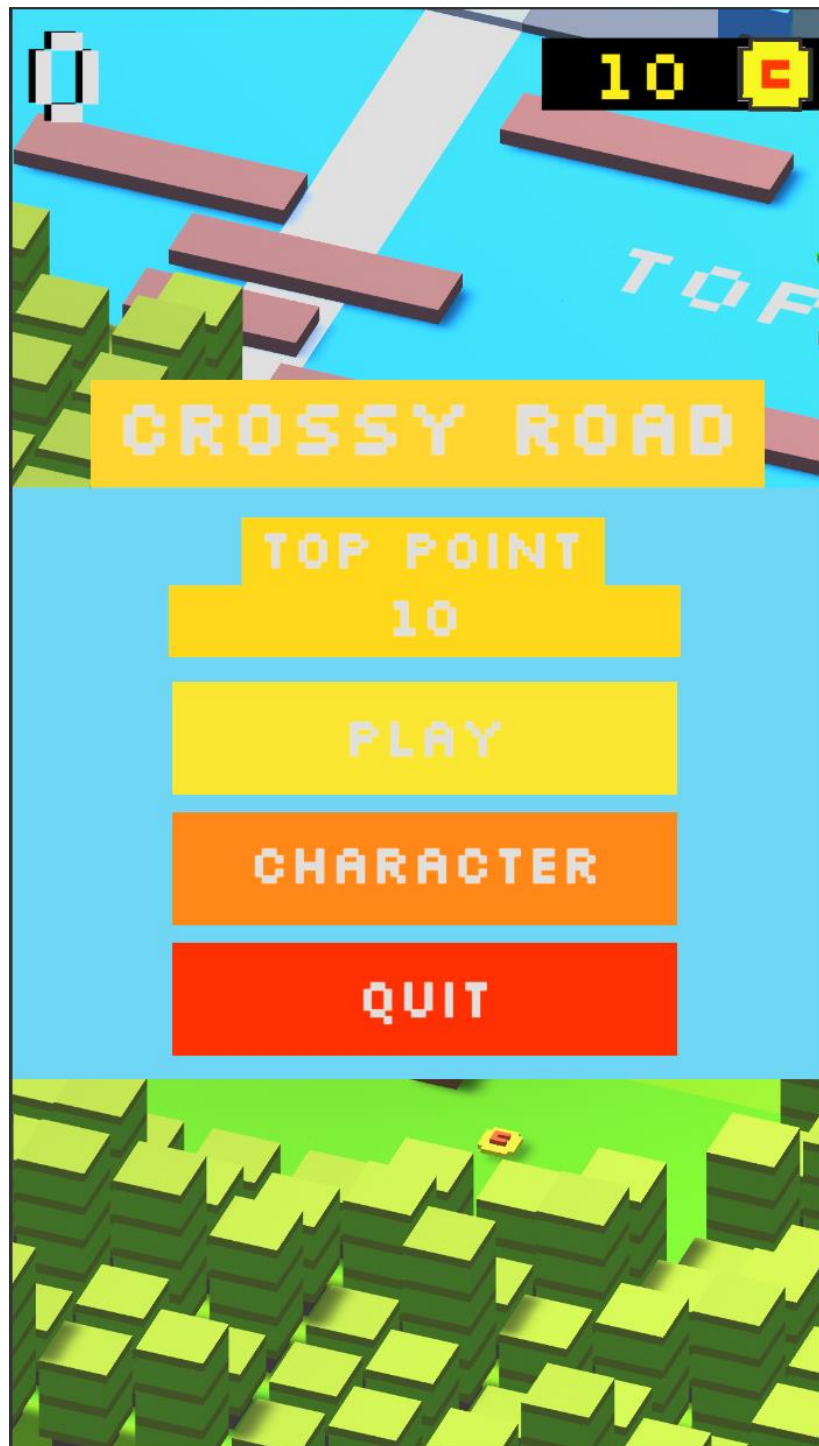
- * Mỗi Character là 1 struct có những thông tin của nhân vật: prefab, trạng thái, giá tiền



Hình 4. 12: ScriptableObject

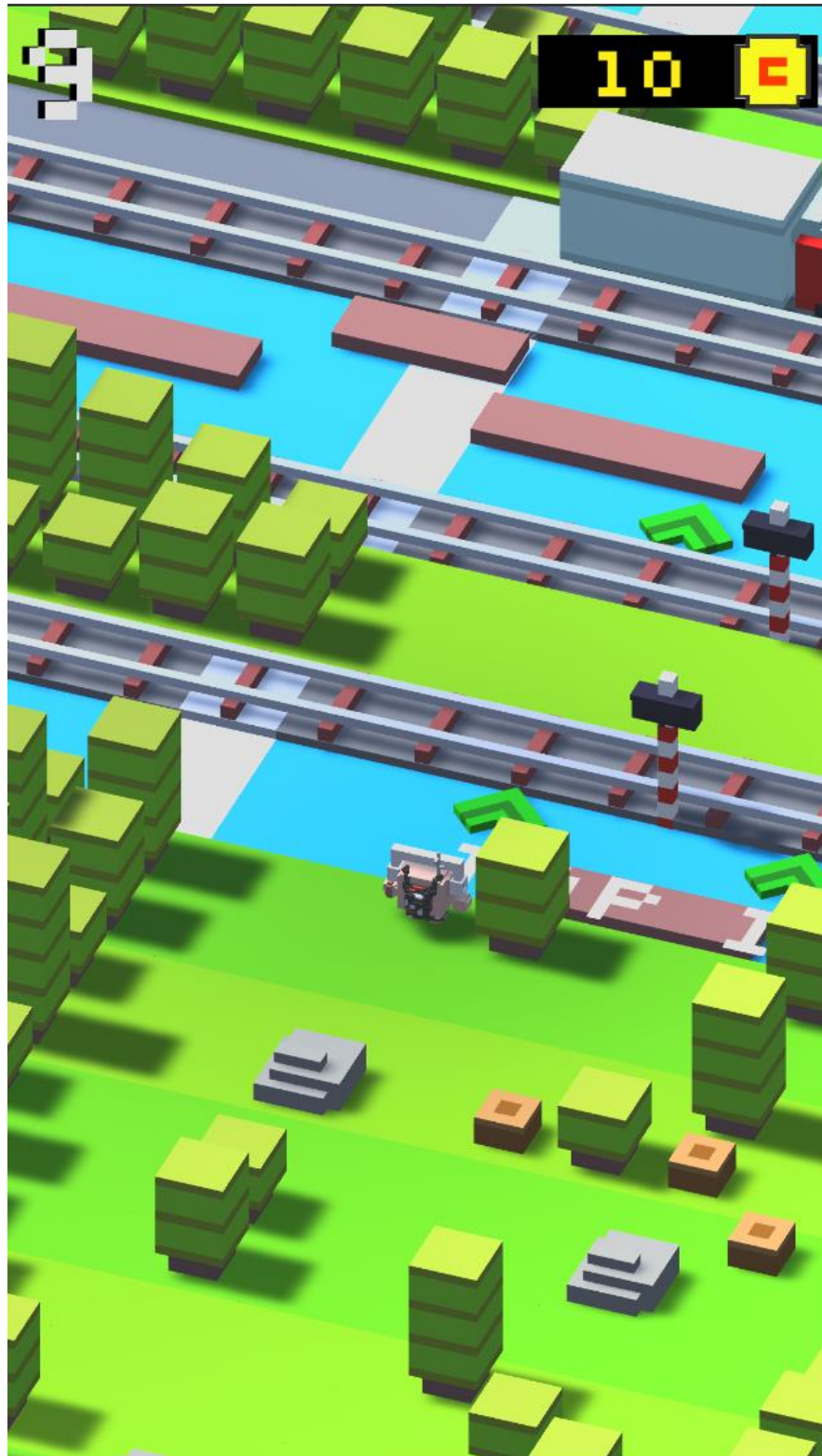
CHƯƠNG 5: XÂY DỰNG GAME VÀ DEMO

Màn hình chính:



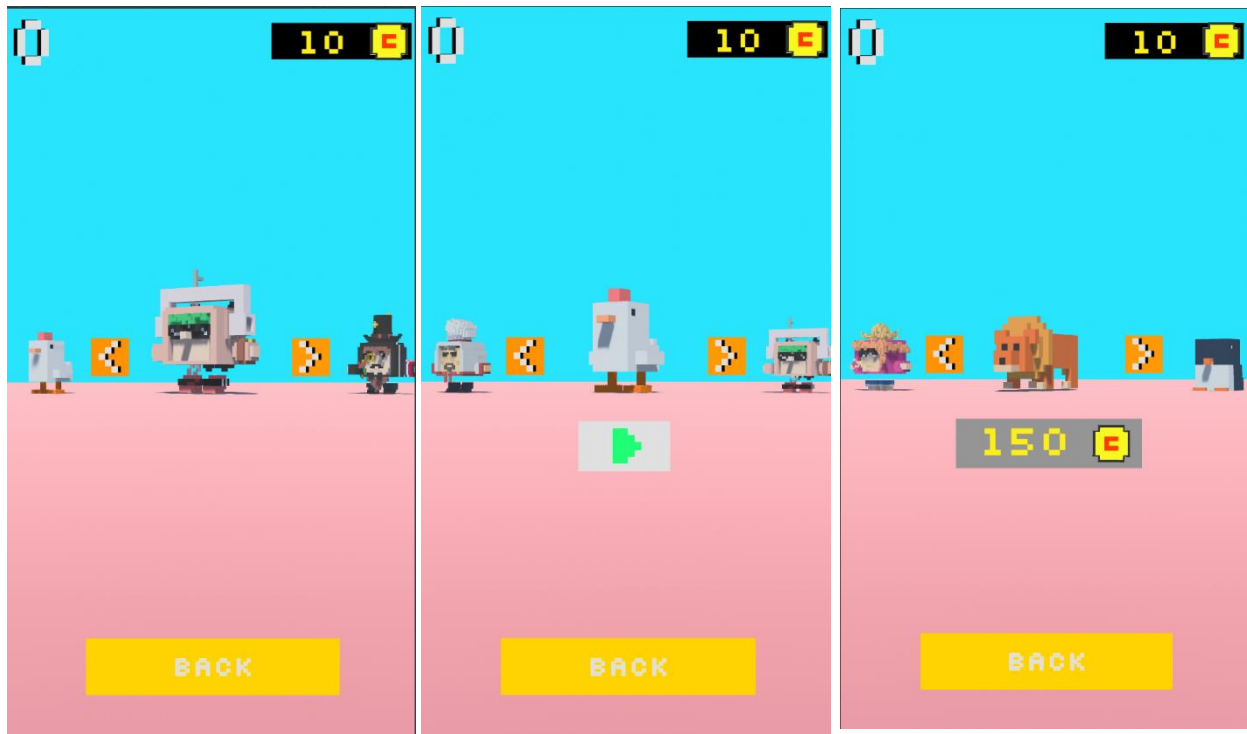
Hình 5.1: Màn hình chính

Màn hình GamePlay:



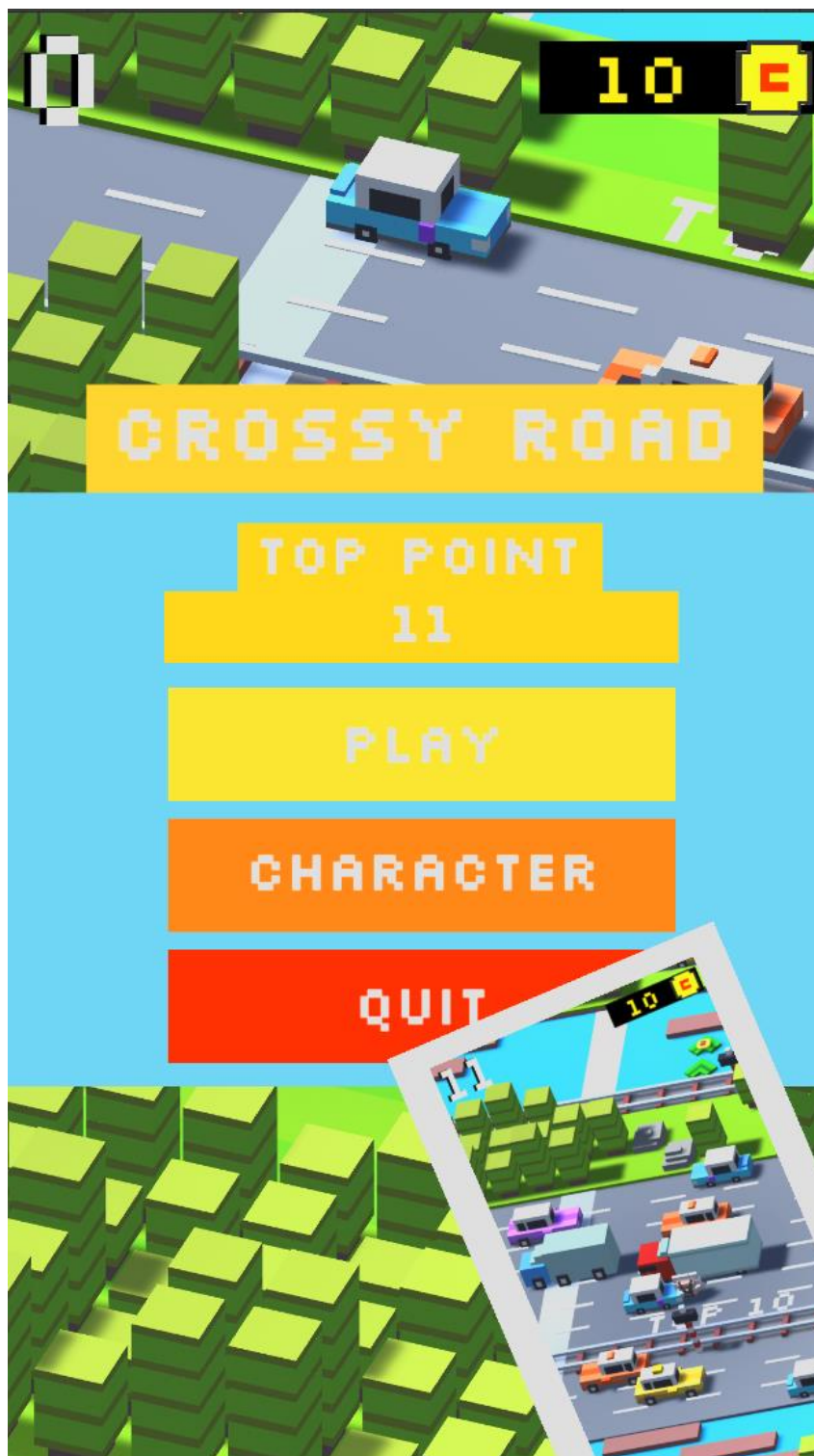
Hình 5.2: Màn hình game play

Màn hình chọn nhân vật:



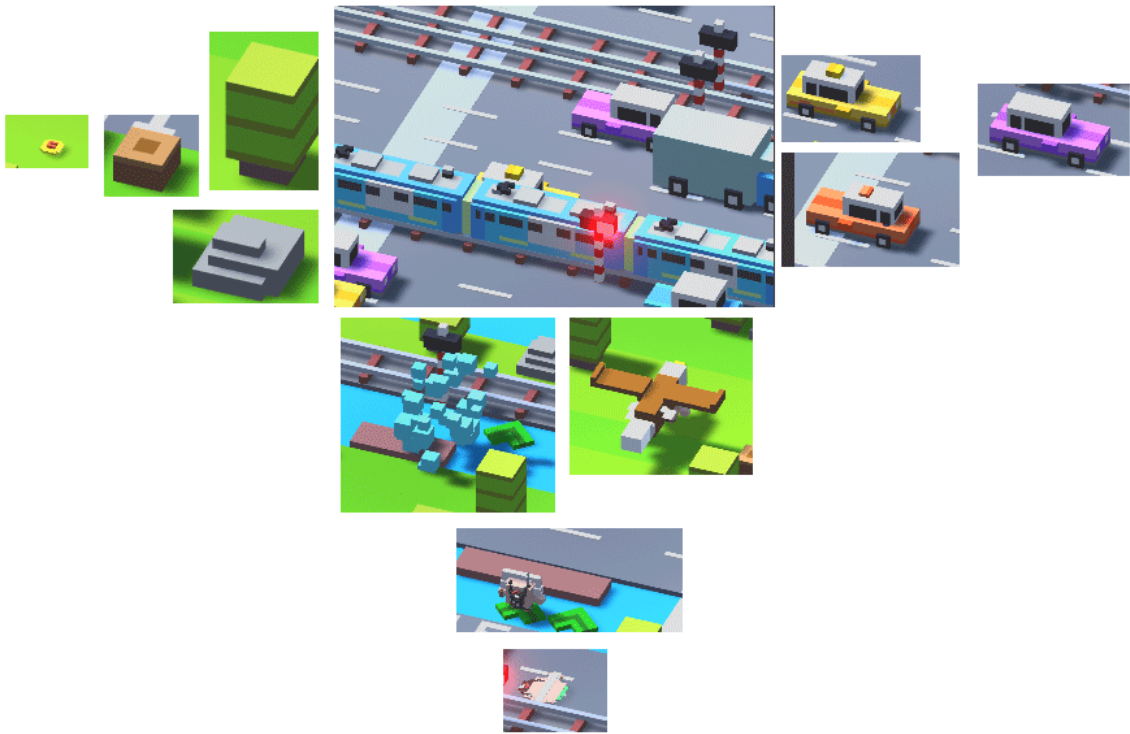
Hình 5.3: Màn hình chọn nhân vật

Khi người chơi đạt được điểm cao nhất



Hình 5.4: Màn hình khi người chơi kết thúc game với điểm cao nhất

Một số hình ảnh trong game:



Hình 5.5: Một số hình ảnh trong game

CHƯƠNG 6: KẾT LUẬN

Trong bài viết này, em đã trình bày về các yếu tố cơ bản của ngành làm game, bao gồm: lịch sử, các công cụ làm game, các ngôn ngữ làm game và các yếu tố làm nên game. Em cũng đã phân tích chi tiết về các yếu tố thiết kế của game Crossy Road : đồ họa, âm thanh, giao diện, điều khiển, nhiệm vụ, map, góc nhìn, hệ thống tiền tệ cũng như hệ thống nhân vật.

Trong quá trình thực hiện đồ án tốt nghiệp, em đã có cơ hội nắm bắt nhiều kiến thức mới về thiết kế và lập trình trò chơi điện tử. Việc này giúp em hiểu rõ hơn về quá trình phát triển và hoàn thiện một trò chơi, từ việc nghiên cứu kiến thức đến việc áp dụng các phương pháp cụ thể. Những kinh nghiệm này không chỉ giúp em hoàn thành đồ án mà còn tạo nên một nền tảng vững chắc, sẵn sàng hỗ trợ công việc tương lai của em trong lĩnh vực này.

Do thời gian có hạn, trình độ và kiến thức của em còn hạn chế nên đồ án này không thể tránh khỏi những thiếu sót. Em rất mong sẽ nhận được sự góp ý của quý thầy cô để đồ án của em được hoàn chỉnh hơn.

Em xin chân thành cảm ơn!

TÀI LIỆU THAM KHẢO

- * https://vi.wikipedia.org/wiki/Crossy_Road
- * https://vi.wikipedia.org/wiki/L%E1%BB%8Bch_s%E1%BB%AD_video_game
- * <https://www.thegioididong.com/game-app/top-11-phan-mem-lap-trinh-game-don-gian-cho-nguoi-moi-bat-dau-1377735>
- * <https://irender.vn/unity-mot-trong-nhung-phan-mem-lam-game-tuyet-voi/>
- * <https://docs.unity3d.com/Manual/index.html/>
- * <http://www.tutorialspoint.com/uml/index.htm>