

**BỘ GIÁO DỤC ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



ĐỒ ÁN TỐT NGHIỆP

PHÁT TRIỂN ỨNG DỤNG TRÒ CHƠI 2D ĐA NỀN TẢNG SỬ DỤNG UNITY

SINH VIÊN THỰC HIỆN: NGUYỄN TẤN NGUYÊN

MÃ SINH VIÊN : 1451020166

KHOA : CÔNG NGHỆ THÔNG TIN

HÀ NỘI - 2024

**BỘ GIÁO DỤC ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



NGUYỄN TẤN NGUYỄN

**PHÁT TRIỂN ỨNG DỤNG TRÒ CHƠI 2D
ĐA NỀN TẢNG SỬ DỤNG UNITY**

CHUYÊN NGÀNH : CÔNG NGHỆ THÔNG TIN
MÃ SỐ : 74.80.201

NGƯỜI HƯỚNG DẪN : ThS. TRẦN THỊ HUỆ

HÀ NỘI - 2024

LỜI CAM ĐOAN

Em xin kính chào thầy/cô,

Em, người viết bản đồ án này, là sinh viên Khoa Công nghệ Thông tin – Trường Đại học Đại Nam. Trong lời cam đoan này, em muốn bày tỏ lòng trân trọng và tôn trọng sâu sắc đến thầy/cô khi đã dành thời gian để đọc và đánh giá bản đồ án của em.

Em cam đoan rằng đồ án này được viết một cách chân thành và trung thực, dựa trên những nghiên cứu và trải nghiệm thực tế của bản thân trong quá trình thực hiện dự án. Tất cả các thông tin, số liệu và ý kiến được trình bày trong đồ án này đều là công sức và hiểu biết của em, không sao chép từ bất kỳ nguồn nào mà không được ghi nhận.

Em cam đoan rằng mọi thông tin trong đồ án này phản ánh một cách chính xác về công việc nghiên cứu của em. Em đã tuân thủ mọi quy định và hướng dẫn của trường và đã làm việc với tinh thần trách nhiệm cao nhất.

Cuối cùng, em xin gửi đến thầy/cô lời cảm ơn chân thành về sự hỗ trợ và động viên trong suốt quá trình thực hiện đồ án này.

Trân trọng,

Hà Nội, ngày 24 tháng 05 năm 2024

Sinh viên thực hiện

Họ và tên sinh viên thực hiện

LỜI CẢM ƠN

Em xin chân thành gửi lời cảm ơn đến cô Trần Thị Huệ và toàn thể cán bộ Khoa Công nghệ Thông tin, Trường Đại học Đại Nam.

Cảm ơn cô đã hướng dẫn và động viên em trong quá trình thực hiện đồ án. Cảm ơn Khoa đã tạo điều kiện thuận lợi cho em hoàn thành dự án này. Em cũng xin gửi lời biết ơn đến Thầy/Cô đã dành thời gian để đọc và đánh giá đồ án của em.

Một lần nữa, em chân thành cảm ơn sự hỗ trợ quý báu từ Thầy/Cô và Khoa trong suốt quá trình này.

Em xin chân thành cảm ơn!

LỜI NÓI ĐẦU

Kính thưa thầy/cô,

Trong thời đại công nghệ ngày nay, việc phát triển trò chơi điện tử không chỉ là một lĩnh vực đầy tiềm năng mà còn là một nghệ thuật kỹ thuật phức tạp. Với sứ mệnh tạo ra một trò chơi 2D thú vị và gây ấn tượng, em đã chọn dự án lập trình game 2D trên nền tảng Unity làm đề tài thực tập của mình.

Trong báo cáo này, em muốn chia sẻ với quý thầy/cô về quá trình phát triển và hoàn thiện dự án của mình. Dự án này không chỉ là một cơ hội để áp dụng và phát triển kỹ năng lập trình của em, mà còn là một hành trình đầy thách thức và học hỏi.

Dự án của em bắt đầu từ việc nghiên cứu và lập kế hoạch chi tiết, qua đó tạo ra một thiết kế game 2D sáng tạo và hấp dẫn. Em đã đối mặt với nhiều thử thách trong việc xây dựng từng phần của trò chơi, từ gameplay đến đồ họa và âm thanh. Quá trình này không chỉ đòi hỏi sự kiên nhẫn và sự sáng tạo mà còn yêu cầu sự hiểu biết sâu rộng về lập trình và công nghệ game.

Trong lời nói đầu này, em sẽ trình bày chi tiết về mục tiêu, phương pháp, kết quả và những bài học quý báu mà em đã học được trong suốt quá trình thực hiện dự án này. Hy vọng rằng, thông qua báo cáo này, em có thể chia sẻ và truyền đạt được những kiến thức và kinh nghiệm mà em đã tích lũy được.

Một lần nữa, xin được gửi lời cảm ơn sâu sắc đến thầy/cô đã luôn hỗ trợ và động viên em trong suốt thời gian làm đề tài này.

Trân trọng,

DANH MỤC BẢNG BIỂU

<i>Bảng 3. 1 Đặc tả use case chức năng cài đặt âm thanh</i>	<i>39</i>
<i>Bảng 3. 2 Đặc tả use case Xem điểm cao</i>	<i>40</i>
<i>Bảng 3. 3 Đặc tả use case xem thành tựu</i>	<i>40</i>
<i>Bảng 3. 4 Đặc tả use case chức năng chơi lại.....</i>	<i>41</i>
<i>Bảng 3. 5 Đặc tả chức năng Thoát trò chơi.....</i>	<i>41</i>
<i>Bảng 3. 6 Đặc tả use case điều khiển nhân vật.....</i>	<i>42</i>
<i>Bảng 3. 7 Test Case.....</i>	<i>48</i>

DANH MỤC HÌNH ẢNH

<i>Hình 2. 1 Unity Editor.....</i>	<i>6</i>
<i>Hình 2. 2 Các thành phần trong Unity Editor</i>	<i>7</i>
<i>Hình 2. 3 Vòng đời của các hàm trong MonoBehaviour</i>	<i>11</i>
<i>Hình 2. 4 Các nền tảng được hỗ trợ bởi Unity</i>	<i>13</i>
<i>Hình 2. 5 Nền tảng Android</i>	<i>14</i>
<i>Hình 2. 6 Nền tảng Windows.....</i>	<i>14</i>
<i>Hình 2. 7 Nền tảng WebGL.....</i>	<i>15</i>
<i>Hình 2. 8 Ngôn ngữ lập trình C#</i>	<i>21</i>
<i>Hình 3. 1 Giao diện Nhiệm vụ thu thập hoa quả</i>	<i>27</i>
<i>Hình 3. 2 Giao diện nhiệm vụ đánh đuổi quái vật</i>	<i>27</i>
<i>Hình 3. 3 Giao diện Cài đặt âm thanh.....</i>	<i>29</i>
<i>Hình 3. 4 Giao diện Game Play.....</i>	<i>29</i>
<i>Hình 3. 5 Giao diện Game Over</i>	<i>30</i>
<i>Hình 3. 6 Giao diện game Complete.....</i>	<i>30</i>
<i>Hình 3. 7 Giao diện Hội thoại.....</i>	<i>31</i>
<i>Hình 3. 8 Biểu đồ Use case tổng quát.....</i>	<i>38</i>
<i>Hình 3. 9 Biểu đồ thành phần Chức năng cài đặt âm thanh</i>	<i>38</i>

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI	1
1.1. Tính cấp thiết của đề tài/ Lý do chọn đề tài	1
1.1.1. Đặt vấn đề	1
1.1.2. Phân tích thị trường	1
1.1.3. Nhu cầu người dùng	1
1.1.4. Ưu điểm cạnh tranh.....	1
1.1.5. Tiềm năng thị trường.....	1
1.1.6. Lý do chọn đề tài	2
1.2. Mục đích nghiên cứu.....	2
1.2.1. Mục tiêu chính.....	2
1.2.2. Mục tiêu cụ thể	2
1.2.3. Đối tượng nghiên cứu.....	2
1.2.4. Kết quả dự kiến.....	2
1.3. Phạm vi nghiên cứu.....	3
1.3.1. Phạm vi hệ thống.....	3
1.3.2. Chức năng cụ thể.....	3
1.3.3. Phạm vi kỹ thuật.....	3
1.3.4. Hạn chế.....	3
1.4. Phương pháp nghiên cứu.....	3
1.4.1. Thiết kế nghiên cứu	3
1.4.2. Phương pháp nghiên cứu	4
1.4.3. Đối tượng nghiên cứu.....	4
1.4.4. Phân tích dữ liệu	4
CHƯƠNG 2. CỞ SỞ LÝ THUYẾT	5
2.1. Tổng quan về Unity Engine	5
2.1.1. Unity là gì?.....	5

2.1.2.	<i>Quá trình phát triển của Unity</i>	5
2.1.3.	<i>Tính năng cơ bản của Unity là gì?</i>	5
2.1.4.	<i>Ưu điểm của Unity</i>	6
2.1.5.	<i>Nhược điểm của Unity</i>	7
2.2.	<i>Tìm hiểu về Unity Engine</i>	7
2.2.1.	<i>Các thành phần trong Unity Editor</i>	7
2.2.2.	<i>Các khái niệm cơ bản trong Unity</i>	9
2.2.3.	<i>Vòng đời của các hàm trong MonoBehaviour</i>	11
2.3.	<i>Các nền tảng được hỗ trợ bởi Unity</i>	13
2.4.	<i>Các thể loại game trên Unity</i>	15
2.4.1.	<i>Các thể loại game hành động</i>	15
2.4.2.	<i>Các thể loại game phiêu lưu hành động</i>	16
2.4.3.	<i>Các thể loại game phiêu lưu</i>	16
2.4.4.	<i>Các thể loại game nhập vai</i>	17
2.4.5.	<i>Các thể loại game mô phỏng</i>	18
2.4.6.	<i>Các thể loại game điện tử chiến thuật</i>	18
2.4.7.	<i>Các thể loại game thể thao</i>	19
2.5.	<i>Một số game nổi tiếng được hoàn thành dựa trên Unity</i>	20
2.6.	<i>Ngôn ngữ lập trình trong môi trường Unity – C Sharp(C#)</i>	21
2.6.1.	<i>C# là gì?</i>	21
2.6.2.	<i>Ưu điểm của C#</i>	22
2.6.3.	<i>Nhược điểm của C#</i>	22
CHƯƠNG 3.	<i>Xây dựng phần mềm</i>	24
3.1.	<i>Đặc tả phần mềm</i>	24
3.1.1.	<i>Game Player</i>	24
3.1.2.	<i>Player Experience</i>	24
3.1.3.	<i>Core Mechanism</i>	24

3.1.4.	<i>Core Loop</i>	24
3.1.5.	<i>Cốt truyện</i>	24
3.2.	<i>Phân tích thiết kế</i>.....	26
3.2.1.	<i>Các tác nhân của game</i>	26
3.2.2.	<i>Các chức năng của game</i>	26
3.2.3.	<i>Các đối tượng game</i>	31
3.2.4.	<i>Biểu đồ Use – case</i>	38
3.2.5.	<i>Biểu đồ hoạt động (Activity Diagram)</i>	42
3.2.6.	<i>Biểu đồ tuần tự (Sequence diagram)</i>	43
3.2.7.	<i>Biểu đồ thành phần (Componment Diagram)</i>	43
3.3.	<i>Kiểm thử</i>.....	44
CHƯƠNG 4.	<i>KẾT QUẢ NGHIÊN CỨU VÀ ĐÁNH GIÁ</i>	49
4.1.	<i>Kết quả đạt được</i>.....	49
4.1.1.	<i>Cơ sở lý thuyết</i>	49
4.1.2.	<i>Cơ sở thực hành</i>	49
4.1.3.	<i>Kỹ năng đạt được</i>	49
4.1.4.	<i>Kết quả ứng dụng</i>	50
4.2.	<i>Đánh giá</i>.....	50
4.2.1.	<i>Ưu điểm</i>	50
4.2.2.	<i>Nhược điểm</i>	50
4.2.3.	<i>Đánh giá chung</i>	50
4.3.	<i>Công việc tương lai</i>.....	50

CHƯƠNG 1.

GIỚI THIỆU ĐỀ TÀI

1.1. Tính cấp thiết của đề tài/ Lý do chọn đề tài

1.1.1. Đặt vấn đề

Trong thời đại công nghệ hiện đại, trò chơi điện tử không chỉ đơn giản là một phương tiện giải trí, mà còn là một phần không thể thiếu của cuộc sống hàng ngày của chúng ta. Tuy nhiên, việc phát triển ứng dụng trò chơi 2D đa nền tảng vẫn đang đối mặt với nhiều thách thức. Với sự gia tăng của các thiết bị di động và sự đa dạng trong hệ điều hành, việc tạo ra một trải nghiệm trò chơi mượt mà và nhất quán trên các nền tảng khác nhau trở thành một mục tiêu khó khăn.

1.1.2. Phân tích thị trường

Thị trường trò chơi di động và trò chơi đa nền tảng đang phát triển mạnh mẽ, với hàng triệu người dùng trên toàn thế giới. Sự tiện lợi của trò chơi di động cũng như tính di động của chúng đã tạo ra một cơ hội lớn cho các nhà phát triển. Tuy nhiên, để nổi bật trong một thị trường cạnh tranh, việc cung cấp một trải nghiệm trò chơi mượt mà và thú vị trên mọi nền tảng là điều bắt buộc.

1.1.3. Nhu cầu người dùng

Người dùng hiện nay đang tìm kiếm những trò chơi mà không chỉ mang lại giây phút giải trí thư giãn, mà còn kích thích trí óc và tương tác xã hội. Ứng dụng trò chơi 2D có thể là một phương tiện tuyệt vời để kết nối con người với nhau thông qua các trải nghiệm chơi game đa người chơi hoặc chế độ trò chơi đa nền tảng.

1.1.4. Ưu điểm cạnh tranh

Dự án của em sẽ tập trung vào việc phát triển một nền tảng trò chơi đa nền tảng linh hoạt và dễ dàng mở rộng. Với sự linh hoạt của Unity Engine, em có thể tối ưu hóa trải nghiệm trò chơi trên nhiều nền tảng mà không cần phải tạo ra các phiên bản riêng biệt cho từng hệ điều hành.

1.1.5. Tiềm năng thị trường

Thị trường trò chơi đang trên đà phát triển mạnh mẽ và với sự gia tăng của các thiết bị di động, em tin rằng việc phát triển một ứng dụng trò chơi đa nền tảng sẽ mang lại tiềm năng lợi nhuận lớn trong tương lai. Sự đa dạng của thị trường và nhu

cầu ngày càng tăng của người dùng cung cấp một cơ hội tuyệt vời để ứng dụng của em phát triển.

1.1.6. Lý do chọn đề tài

Em đã chọn đề tài này vì niềm đam mê với lĩnh vực trò chơi điện tử cũng như nhận thức sâu sắc về tiềm năng của việc phát triển ứng dụng trò chơi 2D đa nền tảng. Đồng thời, em cũng nhận thấy rằng việc thành công trong dự án này không chỉ mang lại lợi ích kinh tế mà còn đóng góp vào sự phát triển của ngành công nghiệp trò chơi điện tử.

1.2. Mục đích nghiên cứu

1.2.1. Mục tiêu chính

Mục tiêu chính của nghiên cứu này là xây dựng và phát triển một ứng dụng trò chơi 2D đa nền tảng sử dụng Unity, với mục đích cung cấp một trải nghiệm chơi game mượt mà và nhất quán trên nhiều thiết bị và hệ điều hành khác nhau.

1.2.2. Mục tiêu cụ thể

- Phát triển một hệ thống trò chơi linh hoạt và dễ dàng mở rộng, cho phép tích hợp và phát triển các tính năng mới một cách hiệu quả.
- Tối ưu hóa trải nghiệm người dùng trên các nền tảng di động và máy tính để bàn, bằng cách đảm bảo tương thích và hiệu suất tốt nhất trên mọi thiết bị.
- Nghiên cứu và áp dụng các phương pháp và công nghệ mới nhằm cải thiện trải nghiệm người dùng và hiệu suất của ứng dụng trò chơi.
- Đánh giá sự hài lòng của người dùng và phản hồi để liên tục cải thiện và tinh chỉnh ứng dụng.

1.2.3. Đối tượng nghiên cứu

Đối tượng nghiên cứu của dự án bao gồm cả những nhà phát triển trò chơi và người chơi, với mục tiêu tạo ra một sản phẩm phù hợp và hấp dẫn đối với cả hai nhóm đối tượng này.

1.2.4. Kết quả dự kiến

Kết quả dự kiến của nghiên cứu này là một ứng dụng trò chơi 2D đa nền tảng hoàn chỉnh và mạnh mẽ, đáp ứng được các tiêu chí về hiệu suất, tương thích và trải nghiệm người dùng. Ngoài ra, dự án cũng mong muốn đóng góp vào việc phát triển

công nghiệp trò chơi điện tử thông qua việc chia sẻ kiến thức và kinh nghiệm thu được trong quá trình nghiên cứu và phát triển.

1.3. Phạm vi nghiên cứu

1.3.1. Phạm vi hệ thống

Dự án sẽ tập trung vào phát triển một hệ thống trò chơi 2D đa nền tảng sử dụng Unity Engine. Phạm vi của dự án bao gồm cả quá trình phát triển, tối ưu hóa và thử nghiệm ứng dụng trên các nền tảng di động (iOS và Android) và máy tính để bàn (Windows và MacOS).

1.3.2. Chức năng cụ thể

Các chức năng cụ thể của ứng dụng trò chơi sẽ bao gồm:

- Cơ chế điều khiển linh hoạt và dễ sử dụng trên mọi thiết bị.
- Hệ thống quản lý cấp độ và tiến trình chơi game.
- Đồ họa 2D đẹp mắt và hiệu ứng đặc sắc.
- Chế độ chơi đơn người và đa người chơi trực tuyến.
- Khả năng tùy chỉnh và mở rộng qua các bản cập nhật sau này.

1.3.3. Phạm vi kỹ thuật

Phạm vi kỹ thuật của dự án bao gồm:

- Sử dụng Unity Engine để phát triển ứng dụng trò chơi.
- Tối ưu hóa hiệu suất để đảm bảo trải nghiệm mượt mà trên mọi nền tảng.
- Tương thích ngược trên các phiên bản hệ điều hành và thiết bị khác nhau.
- Sử dụng ngôn ngữ lập trình C# để viết mã và thực hiện logic của trò chơi.

1.3.4. Hạn chế

Dự án sẽ không đảm bảo tính tương thích với tất cả các thiết bị và phiên bản hệ điều hành. Thay vào đó, phạm vi sẽ tập trung vào việc hỗ trợ các thiết bị phổ biến nhất và các phiên bản hệ điều hành mới nhất.

1.4. Phương pháp nghiên cứu

1.4.1. Thiết kế nghiên cứu

Nghiên cứu sẽ được thiết kế dựa trên một quy trình phát triển ứng dụng trò chơi tiêu chuẩn, bao gồm các giai đoạn sau:

- **Phân tích yêu cầu:** Thu thập và phân tích yêu cầu từ người dùng để xác định các tính năng cần thiết cho ứng dụng trò chơi.
- **Thiết kế:** Xây dựng cấu trúc và giao diện người dùng của ứng dụng, cũng như lập kế hoạch cho việc triển khai các tính năng.
- **Phát triển:** Viết mã và triển khai các tính năng được thiết kế trong giai đoạn trước đó, sử dụng Unity Engine và ngôn ngữ lập trình C#.
- **Kiểm thử:** Thử nghiệm và đánh giá tính năng của ứng dụng trên nhiều thiết bị và nền tảng khác nhau để đảm bảo tính tương thích và hiệu suất.
- **Triển khai:** Phát hành ứng dụng trò chơi cho người dùng cuối thông qua các cửa hàng ứng dụng phổ biến như App Store và Google Play Store.

1.4.2. Phương pháp nghiên cứu

Phương pháp nghiên cứu sẽ bao gồm các bước sau:

- **Nghiên cứu thư viện:** Tiến hành nghiên cứu và phân tích các công nghệ, phương pháp và tiêu chuẩn phát triển trò chơi hiện có để xác định các phương pháp tiếp cận hiệu quả.
- **Thử nghiệm hiệu suất:** Sử dụng công cụ kiểm thử tích hợp của Unity và các công cụ bên ngoài để đo lường và đánh giá hiệu suất của ứng dụng trò chơi trên các thiết bị và nền tảng khác nhau.
- **Thu thập phản hồi:** Sử dụng kỹ thuật thu thập phản hồi từ người dùng thông qua các cuộc thử nghiệm thực tế và bản thử nghiệm để đánh giá sự hài lòng và đề xuất cải tiến.

1.4.3. Đối tượng nghiên cứu

Đối tượng nghiên cứu bao gồm:

- **Nhà phát triển trò chơi:** Đối tượng này sẽ tham gia vào quá trình phát triển và kiểm thử của dự án để cung cấp thông tin và phản hồi cần thiết.
- **Người chơi:** Đối tượng này sẽ tham gia vào các cuộc thử nghiệm và cung cấp phản hồi về trải nghiệm của họ với ứng dụng trò chơi.

1.4.4. Phân tích dữ liệu

Dữ liệu được thu thập từ các cuộc thử nghiệm và phản hồi của người dùng sẽ được phân tích để đánh giá hiệu suất và đề xuất cải tiến cho ứng dụng trò chơi.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về Unity Engine

2.1.1. *Unity là gì?*

Unity là một “cross- platform game engine” tạm hiểu là công cụ phát triển game đa nền tảng được phát triển bởi Unity Technologies. Game engine này được sử dụng để phát triển game trên PC, consoles, thiết bị di động và trên websites. Viết mã trong môi trường Unity (bao gồm cả 2D và 3D) dựa trên ba ngôn ngữ chính: C#, UnityScript và Boo. Trong số này, C# thường là ngôn ngữ phổ biến nhất được sử dụng bởi các nhà phát triển Unity.

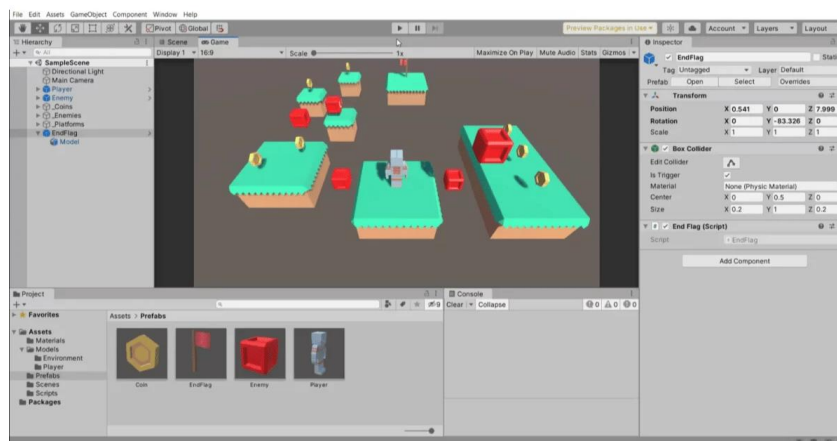
2.1.2. *Quá trình phát triển của Unity*

Ra mắt đầu tiên vào năm 2005 tại sự kiện Apple’s Worldwide Developer Conference bởi nhà sáng lập David Helgason, trải qua hơn 12 năm phát triển, nay Unity đã có version 5.5 hoàn thiện hơn về rất nhiều mặt. Tháng 5-2012 theo cuộc khảo sát Game Developer Megazine được công nhận là Game engine tốt nhất cho mobile. Năm 2014 Unity thắng giải “Best Engine” tại giải UK’s annual Develop Industry Excellence.

2.1.3. *Tính năng cơ bản của Unity là gì?*

Unity hỗ trợ mạnh các tính năng sau:

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



Hình 2. 1 Unity Editor¹

2.1.4. Ưu điểm của Unity

- Đa nền tảng: Unity cho phép phát triển ứng dụng và trò chơi cho nhiều nền tảng khác nhau như iOS, Android, PC, Mac, các hệ máy chơi game console và thậm chí là các thiết bị thực tế ảo và thực tế tăng cường.
- Cộng đồng lớn: Unity có một cộng đồng rộng lớn, với hàng nghìn người dùng, diễn đàn, blog và tài liệu hướng dẫn trực tuyến. Điều này giúp người dùng dễ dàng tìm kiếm thông tin, hỗ trợ và tài nguyên.
- Dễ học: Unity cung cấp một giao diện người dùng thân thiện và dễ sử dụng, giúp người mới bắt đầu dễ dàng học và sử dụng công cụ này.
- Cộng tác đa người: Unity hỗ trợ cộng tác đa người, cho phép nhiều nhà phát triển làm việc cùng nhau trên cùng một dự án.
- Tiện ích và plugin: Unity có một hệ sinh thái plugin phong phú, cung cấp các tiện ích và plugin để mở rộng chức năng của nền tảng.

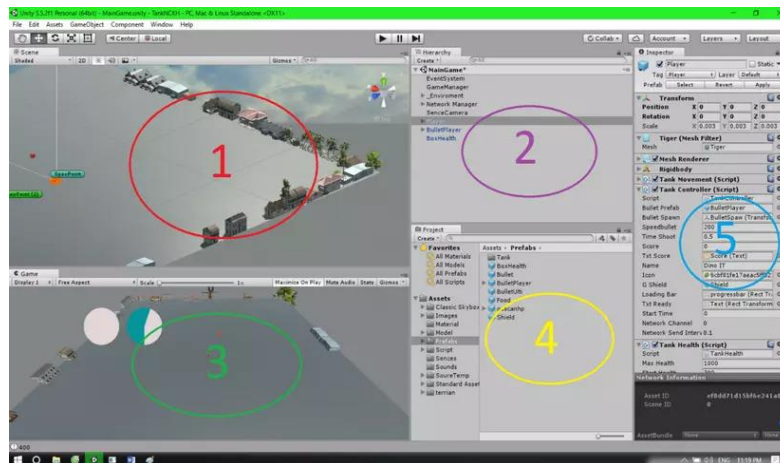
¹[https://www.bkns.vn/unity-la-gi.html#:~:text=C%C3%A1c%20t%C3%ADnh%20%C4%83ng%20%C6%A1%20b%E1%BA%A3n%20c%E1%BB%A7a%20Unity,-Ph%E1%BA%A7n%20m%E1%BB%81m%20Unity&text=Hi%E1%BB%83n%20th%E1%BB%8B%20m%C3%B4%20h%C3%ACnh%20d%E1%BA%A1ng,%E2%80%93%20VR%20\(Virtual%20reality\).](https://www.bkns.vn/unity-la-gi.html#:~:text=C%C3%A1c%20t%C3%ADnh%20%C4%83ng%20%C6%A1%20b%E1%BA%A3n%20c%E1%BB%A7a%20Unity,-Ph%E1%BA%A7n%20m%E1%BB%81m%20Unity&text=Hi%E1%BB%83n%20th%E1%BB%8B%20m%C3%B4%20h%C3%ACnh%20d%E1%BA%A1ng,%E2%80%93%20VR%20(Virtual%20reality).)

2.1.5. Nhược điểm của Unity

- Hiệu suất: Mặc dù Unity có khả năng tối ưu hóa, nhưng việc phát triển ứng dụng phức tạp có thể gặp vấn đề về hiệu suất, đặc biệt là trên các thiết bị di động yếu.
- Có thể nặng nề: Unity có thể trở nên nặng nề và khó quản lý với các dự án lớn, đặc biệt là khi có nhiều người làm việc cùng một lúc.
- Giá cả: Unity cung cấp phiên bản miễn phí và các phiên bản trả phí với nhiều tính năng mở rộng. Tuy nhiên, việc sử dụng các tính năng cao cấp có thể đòi hỏi phải trả phí.
- Có thể hạn chế trong một số trường hợp: Mặc dù Unity là một công cụ mạnh mẽ, nhưng có thể có những trường hợp nơi nó không phù hợp cho một số yêu cầu cụ thể của dự án.

2.2. Tìm hiểu về Unity Engine

2.2.1. Các thành phần trong Unity Editor



Hình 2. 2 Các thành phần trong Unity Editor²

1. Cửa sổ Scenes

- Phần này phân hiển thị các đối tượng trong scenes một cách trực quan, có thể lựa chọn các đối tượng, kéo thả, phóng to, thu nhỏ, xoay các đối tượng ...

² <https://viblo.asia/p/gioi-thieu-ve-unity-engine-game-engine-pho-bien-nhat-hien-nay-V3m5WBj8lO7>

- Phần này có để thiết lập một số thông số như hiển thị ánh sáng, âm thanh, cách nhìn 2D hay 3D ... Khung nhìn Scene là nơi bố trí các Game Object như cây cối, cảnh quan, enemy, player, camera, ... trong game. Sự bố trí hoạt cảnh là một trong những chức năng quan trọng nhất của Unity.

2. Cửa sổ Hierarchy

- Tab hierarchy là nơi hiển thị các Game Object trong Scenes hiện hành. Khi các đối tượng được thêm hoặc xóa trong Scenes, tương ứng với các đối tượng đó trong cửa sổ Hierarchy.
- Tương tự trong tab Project, Hierarchy cũng có một thanh tìm kiếm giúp quản lý và thao tác với các Game Object hiệu quả hơn đặc biệt là với các dự án lớn.

3. Cửa sổ game

- Đây là màn hình demo Game, là góc nhìn từ camera trong game.
- Thanh công cụ trong cửa sổ game cung cấp các tùy chỉnh về độ phân giải màn hình, thông số (stats), gizmos, tùy chọn bật tắt các component...

4. Cửa sổ Project

- Đây là cửa sổ explorer của Unity, hiển thị thông tin của tất cả các tài nguyên (Assets) trong game của bạn.
- Cột bên trái hiển thị assets và các mục yêu thích dưới dạng cây thư mục tương tự như Windows Explorer. Khi click vào một nhánh trên cây thư mục thì toàn bộ nội dung của nhánh đó sẽ được hiển thị ở khung bên phải. Ta có thể tạo ra các thư mục mới bằng cách Right click -> Create -> Folder hoặc nhấn vào nút Create ở góc trên bên trái cửa sổ Project và chọn Folder. Các tài nguyên trong game cũng có thể được tạo ra bằng cách này.
- Phía trên cây thư mục là mục Favorites, giúp chúng ta truy cập nhanh vào những tài nguyên thường sử dụng. Chúng ta có thể đưa các tài nguyên vào Favorites bằng thao tác kéo thả.
- Đường dẫn của thư mục tài nguyên hiện tại. Chúng ta có thể dễ dàng tiếp cận các thư mục con hoặc thư mục gốc bằng cách click chuột vào mũi tên hoặc tên thư mục.

5. Cửa sổ Inspector

- Cửa sổ Inspector hiển thị chi tiết các thông tin về Game Object đang làm việc, kể cả những component được đính kèm và thuộc tính của nó. Bạn có thể điều

chỉnh, thiết lập mọi thông số và chức năng của Game Object thông qua cửa sổ Inspector.

- Mọi thuộc tính thể hiện trong Inspector đều có thể dễ dàng tùy chỉnh trực tiếp mà không cần thông qua một kịch bản định trước. Tuy nhiên Scripting API cung cấp một số lượng nhiều và đầy đủ hơn do giao diện Inspector là có giới hạn.
- Các thiết lập của từng component được đặt trong menu. Các bạn có thể click chuột phải, hoặc chọn icon hình bánh răng nhỏ để xuất hiện menu.
- Ngoài ra Inspector cũng thể hiện mọi thông số Import Setting của asset đang làm việc như hiển thị mã nguồn của Script, các thông số animation, ...

2.2.2. Các khái niệm cơ bản trong Unity

1. GameObject

Một đối tượng cụ thể trong game gọi là một game object, có thể là nhân vật, đồ vật nào đó. Ví dụ: cây cối, xe cộ, nhà cửa, người...

2. Component

Một GameObject sẽ có nhiều thành phần cấu tạo nên nó như là hình ảnh (sprite render), tập hợp các hành động (animator), thành phần xử lý va chạm (collision), tính toán vật lý (physical), mã điều khiển (script), các thành phần khác... mỗi thứ như vậy gọi là một component của GameObject.

3. Sprite

Là một hình ảnh 2D của một game object có thể là hình ảnh đầy đủ, hoặc có thể là một bộ phận nào đó.

4. Animation

Là tập một hình ảnh động dựa trên sự thay đổi liên tục của nhiều sprite khác nhau.

5. Key Frame

Key Frame hay Frame là một trạng thái của một animation. Có thể được tạo nên từ 1 sprite hay nhiều sprite khác nhau.

6. Prefabs

Là một khái niệm trong Unity, dùng để sử dụng lại các đối tượng giống nhau có trong game mà chỉ cần khởi tạo lại các giá trị vị trí, tỉ lệ biến dạng và góc quay

từ một đối tượng ban đầu. Ví dụ: Các đối tượng là đồng tiền trong game Mario đều có xử lý giống nhau, nên ta chỉ việc tạo ra một đối tượng ban đầu, các đồng tiền còn lại sẽ sử dụng prefabs. Hoặc khi ta lát gạch cho một cái nền nhà, các viên gạch cũng được sử dụng là prefabs.

7. Sounds

Âm thanh trong game.

8. Script

Script là tập tin chứa các đoạn mã nguồn, dùng để khởi tạo và xử lý các đối tượng trong game. Trong Unity có thể dùng C#, Java Script, BOO để lập trình Script.

9. Scenes

Quản lý tất cả các đối tượng trong một màn chơi của game.

10. Assets

Bao gồm tất cả những gì phục vụ cho dự án game như sprite, animation, sound, script, scenes...

11. Camera

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

12. Transform

Là 3 phép biến đổi tịnh tiến, quay theo các trục, và phóng to thu nhỏ một đối tượng

2.2.3. Vòng đời của các hàm trong Monobehaviour

ngay sau hàm Awake của nó.

- Start(): hàm Start() được gọi sau OnEnable, trước khi các frames bắt đầu chạy hay trước các hàm Update. Cũng như Awake, nó chỉ được gọi một lần duy nhất.

Như vậy, các hàm thuộc loại Initialization thường được sử dụng để khởi tạo các giá trị có sẵn hoặc cache các components (GetComponent), trong đó chỉ có OnEnable() có thể được gọi nhiều lần

2. Game Logic

- Update(): chắc hẳn hàm này đã quá quen thuộc khi chúng ta mới bắt đầu sử dụng Unity engine, Update được gọi mỗi lần vào mỗi frame và được xem như hàm xử lý chính của vòng lặp game.
- LateUpdate(): LateUpdate() được gọi như Update, mỗi frame một lần, điểm khác biệt duy nhất là nó được gọi sau khi các hàm Update của tất cả các MonoBehaviour đã được thực thi xong.

3. Physics

- FixedUpdate: Hàm này được gọi không phụ thuộc vào vòng lặp chính của game mà gọi theo vòng lặp của vật lý trong game, thường nó được gọi cố định vào mỗi 0.02s theo mặc định của project setting, con số này gọi là Fixed Timestep hay fixedDeltaTime.

Vì vậy FixedUpdate có thể được gọi nhiều hơn một lần hoặc không gọi trong 1 frame tùy theo FPS của game

- Internal Physics Update: Đây là vùng xử lý chính của Physics System trong unity được xử lý riêng mà chúng ta không can thiệp, với một số chức năng sẵn có như áp trọng lực, trigger các hàm va chạm, di chuyển các vật thể sử dụng vật lý (rigidbody, joints,...)

Vùng này được thực thi sau FixedUpdate, vì vậy các xử lý liên quan tới vật lý thì chúng ta nên đặt ở trong FixedUpdate thay vì Update để Physics System của Unity có thể cập nhật vị trí và xét va chạm kịp thời.

4. Decommissioning

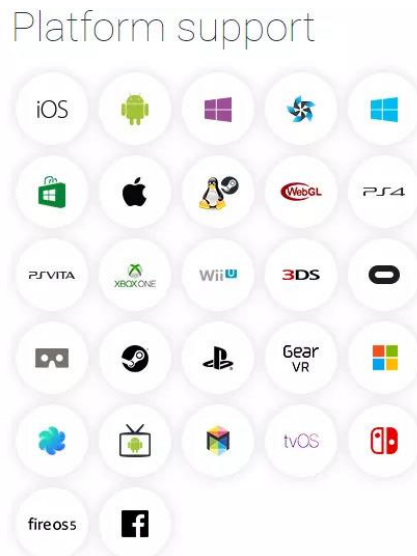
OnApplicationQuit(): cái tên cũng nói lên được công dụng của hàm này rồi, nó sẽ được gọi trước khi thoát game (hoặc tắt Play mode trong Unity Editor)

OnDisable(): hàm này tương ứng với OnEnable mà mình đã đề cập ở trên, có thể được gọi nhiều lần nếu GameObject active-deactive nhiều lần (hoặc enable-disable Component gắn vào GameObject)

OnDestroy(): cuối cùng là OnDestroy, hàm này sẽ được gọi ở cuối frame, đồng

nghĩa với việc khi xóa một GameObject trong scene (bằng script hoặc các sự kiện như thoát game, chuyển scene,...) hay sử dụng hàm Destroy(gameObject), nó sẽ không xóa ngay lập tức mà phải đến cuối frame đó nó mới được dọn dẹp.

2.3. Các nền tảng được hỗ trợ bởi Unity



Hình 2. 4 Các nền tảng được hỗ trợ bởi Unity⁴

Unity có mặt trên hầu như mọi nền tảng từ Windows, Macs, iOS, Android, Windows Phone cho đến Steam, Playstation, Xbox, Wii U. Các ứng dụng game đã và đang phát triển trên rất nhiều nền tảng khác nhau và vì vậy Unity cũng đang phát triển theo xu hướng này. Việc chuyển đổi 1 game từ nền tảng này sang nền tảng khác trước đây đòi hỏi một số lượng công việc cực kì lớn cùng sự kết hợp với đội ngũ nhân viên outsourcing ngoài công ty và phải mất đến vài tháng để thực hiện việc chuyển đổi này. Với Unity, việc xuất/nhập sang 1 nền tảng khác trở nên dễ dàng hơn rất nhiều.

Android là hệ điều hành trên điện thoại di động (và hiện nay là cả trên một số đầu phát HD, HD Player, TV) phát triển bởi Google và dựa trên nền tảng Linux. Trước đây, Android được phát triển bởi công ty liên hợp Android (sau đó được Google mua lại vào năm 2005).

⁴ <https://viblo.asia/p/gioi-thieu-ve-unity-engine-game-engine-pho-bien-nhat-hien-nay-V3m5WBj8lO7>



Hình 2. 5 Nền tảng Android⁵

Windows là hệ điều hành trên máy tính hoặc laptop phát triển bởi công ty phần mềm nổi tiếng Microsoft tạo ra. Windows là một nền tảng hệ điều hành mạnh mẽ và linh hoạt, phù hợp với nhiều đối tượng người dùng từ cá nhân đến doanh nghiệp. Với lịch sử phát triển lâu dài và liên tục cải tiến, Windows tiếp tục là một trong những hệ điều hành phổ biến và được ưa chuộng nhất trên thế giới.



Hình 2. 6 Nền tảng Windows⁶

WebGL (Web Graphics Library) là một thư viện đồ họa dành cho web, nó được dẫn xuất từ OpenGL ES (thư viện đồ họa 2D và 3D trên hệ thống nhúng: điện thoại, đồ điện tử, xe cơ giới). WebGL cung cấp các chức năng cơ bản tương tự OpenGL ES và hoạt động tốt trên các phân cứng đồ họa 3D hiện đại. WebGL là javascript API có thể sử dụng được trong HTML5, WebGL được code trong tag <canvas> của HTML 5, điều này cho phép trình duyệt có thể truy cập và sử dụng GPU để xuất ra

⁵ <https://fonesmart.com.vn/tin-cong-nghe/he-dieu-hanh-android-la-gi-tim-hieu-chi-tiet-ve-he-dieu-hanh-android-va-nhung-thuong-hieu-dien-thoai-chay-android-noi-tieng-n779.html>

⁶ <https://www.dienmayxanh.com/kinh-nghiem-hay/he-dieu-hanh-windows-la-gi-596407>

card đồ họa. WebGL được hỗ trợ bởi đa số các trình duyệt hiện đại: Chrome, FireFox, IE, Opera...



Hình 2. 7 Nền tảng WebGL⁷

2.4. Các thể loại game trên Unity

2.4.1. Các thể loại game hành động

1. Thể loại game đi cảnh - Platform games

Đây là thể loại game tập trung vào việc nhảy qua các nền tảng và vượt qua các chướng ngại vật để hoàn thành màn chơi. Các nhân vật chính thường là những anh hùng phiêu lưu, nhân vật hoạt hình dễ thương hoặc nhân vật từ các bộ phim nổi tiếng.

2. Thể loại game bắn súng - Shooter games

Là một trong những thể loại game phổ biến nhất, shooter games yêu cầu người chơi sử dụng vũ khí để chiến đấu và tiêu diệt đối thủ. Thể loại này bao gồm các trò chơi từ góc nhìn thứ nhất (First-Person Shooter - FPS) và góc nhìn thứ ba (Third Person Shooter - TPS).

3. Thể loại game đối kháng - Fighting games

Trở thành võ sĩ mạnh mẽ và tham gia vào các trận chiến võ thuật độc đáo và hấp dẫn. Các nhân vật trong fighting games thường có các kỹ năng và đòn tấn công đặc biệt riêng.

4. Thể loại game hành động lén lút - Stealth game

⁷ <https://bizfly.vn/techblog/webgl-la-gi.html>

Thể loại này đòi hỏi người chơi tinh tế và nguy trang để vượt qua các vùng đất nguy hiểm mà không bị phát hiện. Các nhiệm vụ thường liên quan đến ẩn náu, đánh lừa kẻ thù và tiêu diệt chúng một cách thâm lặng.

5. Thể loại game sinh tồn - Survival games

Thể loại này đặt người chơi vào môi trường đầy nguy hiểm và giới hạn tài nguyên. Người chơi phải tìm cách sinh tồn và xây dựng căn cứ để chống lại những thách thức đáng sợ.

6. Thể loại game âm nhạc giai điệu - Rhythm games

Người chơi phải điều khiển nhân vật theo nhịp điệu nhạc để hoàn thành các màn chơi. Thể loại này thường yêu cầu khả năng điều khiển chính xác và đồng bộ âm nhạc.

7. Thể loại game bắn súng sinh tồn - Battle Royale games

Lấy cảm hứng từ các bộ phim và tiểu thuyết cùng tên, thể loại này đưa người chơi vào một trận chiến sinh tử, nơi chỉ có một người sống sót cuối cùng.

2.4.2. Các thể loại game phiêu lưu hành động

1. Thể loại game kinh dị sinh tồn - Survival horror

Ghép nối yếu tố kinh dị và sinh tồn, người chơi sẽ phải đối mặt với những tình huống đáng sợ và tìm cách sinh tồn. Những âm thanh đáng sợ và môi trường u ám thường làm nổi bật thêm không khí kinh dị trong trò chơi.

2. Thể loại game hành động phiêu lưu - Metroidvania

Thể loại này tập trung vào việc khám phá và mở khóa các khu vực mới trong một thế giới rộng lớn. Người chơi cần phải giải quyết các câu đố và đánh bại các đối thủ để tiến tới mục tiêu cuối cùng.

2.4.3. Các thể loại game phiêu lưu

1. Thể loại game phiêu lưu văn bản - Text adventures

Sử dụng văn bản và câu chuyện để tạo ra trải nghiệm phiêu lưu tương tác. Người chơi cần phải đọc và hiểu câu chuyện để đưa ra các quyết định phù hợp.

2. Thể loại game tiểu thuyết tương tác - Visual novels

Tương tự như Text adventures nhưng có phần lớn nội dung dựa vào câu chuyện và hình ảnh động.

3. Thể loại game điện ảnh tương tác - Interactive movie

Thể loại này mang đến cho người chơi trải nghiệm giống như xem một bộ phim tương tác. Người chơi tham gia vào câu chuyện bằng cách lựa chọn hành động và quyết định trong các cảnh phim, ảnh hưởng đến kết quả của câu chuyện.

4. Thể loại game phiêu lưu 3D thời gian thực - Real-time 3D adventures

Đưa người chơi vào những thế giới 3D tuyệt đẹp và tươi đẹp, thể loại này tập trung vào khám phá môi trường và giải quyết các câu đố. Trò chơi đa dạng và kịch tính với các cảnh hành động và môi trường sống động.

2.4.4. Các thể loại game nhập vai

1. Thể loại nhập vai hành động - Action RPG

Kết hợp yếu tố hành động và nhập vai, thể loại này cho phép người chơi tiến hành những cuộc phiêu lưu đầy hành động và chặt chém, trong đó có sự phát triển nhân vật và nâng cấp kỹ năng.

2. Thể loại game nhập vai trực tuyến - MMORPG

Thể loại game này cho phép hàng nghìn người chơi tham gia vào một thế giới ảo chung và tương tác với nhau thông qua mạng internet. Người chơi có thể tham gia vào các cuộc chiến đội, thực hiện nhiệm vụ và xây dựng nhân vật của riêng mình.

3. Thể loại game nhập vai dạng lưới - Roguelikes

Thể loại này xuất phát từ các trò chơi nhập vai cổ điển với cấu trúc lưới 2D. Các trận chiến và môi trường trong game được sinh tồn một cách ngẫu nhiên, đòi hỏi người chơi phải tái xây dựng và tăng cường nhân vật của mình sau mỗi lần chơi lại.

4. Thể loại game nhập vai chiến lược - Tactical RPG

Thể loại này tập trung vào yếu tố chiến thuật và đòi hỏi người chơi lựa chọn và triển khai các chiến lược chiến đấu trong các trận đánh hấp dẫn.

5. Thể loại game nhập vai thế giới mở - Sandbox RPG

Đặt người chơi vào một thế giới mở rộng, thể loại này cho phép người chơi tự do tạo dựng cuộc sống, lựa chọn nhiệm vụ và tham gia vào các hoạt động tùy thích.

6. Thể loại game nhập vai nhóm theo lượt - First-person party-based RPG

Thể loại này kết hợp yếu tố nhập vai nhóm với góc nhìn thứ nhất. Người chơi sẽ điều khiển một nhóm nhân vật trong cuộc phiêu lưu và cùng nhau thực hiện các nhiệm vụ và trận đấu.

7. Thể loại game nhập vai Nhật Bản - JRPG

Xuất phát từ Nhật Bản, thể loại này có những yếu tố đặc trưng như phong cách nghệ thuật manga và anime, hệ thống chiến đấu theo lượt và cốt truyện phong phú.

8. Thể loại game nhập vai thu phục quái vật - Monster Tamer

Đặc biệt phổ biến ở Nhật Bản, thể loại này cho phép người chơi thu phục và huấn luyện các loài quái vật để tham gia vào các trận đấu và cuộc phiêu lưu.

2.4.5. Các thể loại game mô phỏng

1. Thể loại game mô phỏng xây dựng, quản lý - Construction and management simulation

Thể loại này cho phép người chơi xây dựng và quản lý một hệ thống hoạt động như thành phố, công viên giải trí, trang trại, hoặc công ty.

2. Thể loại game mô phỏng cuộc sống - Life simulation

Người chơi có thể giả lập cuộc sống hàng ngày của nhân vật trong trò chơi, bao gồm sinh hoạt cá nhân, công việc, và giao tiếp xã hội.

3. Thể loại game mô phỏng phương tiện - Vehicle simulation

Thể loại này cho phép người chơi cảm nhận việc điều khiển các loại phương tiện như xe hơi, máy bay, tàu hỏa hoặc tàu thủy trong môi trường 3D sống động.

2.4.6. Các thể loại game điện tử chiến thuật

1. 4X game

Đặc trưng cho Explore (khám phá), Expand (mở rộng), Exploit (khai thác) và Exterminate (tiêu diệt), thể loại này tập trung vào xây dựng đế chế và chiến tranh quy mô lớn.

2. Thể loại game pháo binh - Artillery game

Trò chơi chiến thuật tương tác giữa các đối thủ, trong đó người chơi cần phải tính toán góc độ và sức mạnh để tiêu diệt đối thủ bằng vũ khí pha pháo.

3. Auto chess - Auto battler

Người chơi tạo ra và điều chỉnh một đội hình chiến đấu và sau đó quan sát chúng tự động chiến đấu với đối thủ. Trò chơi này yêu cầu người chơi có sự chiến thuật trong việc xây dựng đội hình và quản lý tài nguyên.

4. Thể loại game đấu trường trực tuyến đa người chơi - Multiplayer online battle arena

Thể loại này kết hợp yếu tố chiến thuật và đấu trường đa người chơi trực tuyến. Người chơi cạnh tranh trong các trận đấu đối kháng với đội hình riêng.

5. Thể loại game chiến thuật thời gian thực - Real-time strategy

Trong thể loại này, người chơi đưa ra quyết định chiến thuật và điều khiển quân đội một cách trực tiếp trong thời gian thực.

6. Chiến lược thời gian thực - Real-time tactics

Tương tự như Real-time strategy, nhưng tập trung vào việc điều khiển một nhóm nhân vật hoặc đơn vị chiến đấu một cách chi tiết hơn.

7. Thể loại game tháp phòng thủ - Tower defense

Người chơi xây dựng các cấu trúc phòng thủ để ngăn chặn quái vật hoặc kẻ thù tấn công vào điểm mục tiêu chính.

8. Thể loại game chiến thuật theo lượt - Turn-based strategy

Thể loại này cho phép người chơi thực hiện các hành động và chiến thuật trong các lượt đấu liên tục.

9. Thể loại game đại chiến lược - Grand strategy wargame

Thể loại này tập trung vào mô phỏng các cuộc chiến tranh quy mô lớn và quản lý các khía cạnh lớn của một đế chế.

2.4.7. Các thể loại game thể thao

1. Thể loại game đua xe - Racing

Thể loại này tập trung vào các trò chơi đua xe thú vị và đa dạng, từ đua xe F1, đua xe địa hình đến đua xe đường phố.

2. Thể loại game thể thao - Sports game

Tập hợp các trò chơi mô phỏng các môn thể thao phổ biến như bóng đá, bóng rổ, bóng chày và nhiều môn khác.

3. Thể loại game thể thao thi đấu - Competitive

Người chơi tham gia vào các cuộc thi thể thao hoặc sự kiện đấu đỉnh cao và cạnh tranh với nhau để đạt được chiến thắng.

4. Thể loại game đấu võ - Sports-based fighting

Kết hợp yếu tố thể thao và đối kháng, thể loại này cho phép người chơi tham gia vào các trận đấu võ thuật thú vị.

2.5. Một số game nổi tiếng được hoàn thành dựa trên Unity

1. Rust

Ra mắt vào năm 2013, Rust được phát triển bởi studio Facepunch và thuộc thể loại phiêu lưu hành động. Lấy cảm hứng từ Minecraft, DayZ và Stalker, trò chơi là một trò chơi sinh tồn mới với môi trường khắc nghiệt nơi cuộc sống con người có thể trở nên mong manh hơn bao giờ hết. Chiến thắng trong Rust đòi hỏi bạn phải tiếp tục chiến đấu để tồn tại. Rust được xây dựng trên Engine Unity 3D Game 5.0, đó là lý do tại sao nó khá hoàn chỉnh về mặt đồ họa.

2. Touch

Chuyển động của nhân vật cực kỳ mượt mà, tóc và quần áo được tạo hình tinh xảo, hiệu ứng va chạm chân thực, kết hợp với các công nghệ phức tạp như chiếu sáng và tô màu đồng thời...Không chỉ vậy, Touch còn có khả năng hoạt động đa nền tảng tốt, không có nhiều sự khác biệt về màn hình của các phiên bản khác nhau dù là PC hay di động.

Đối với trò chơi, Touch cung cấp một mô hình thể loại âm nhạc và khiêu vũ hoàn toàn mới. Khi tham gia trò chơi, bạn phải chọn trong 3 chế độ chơi chính: Chạm, Bong bóng và Đấu. Ngoài ra, Touch còn cung cấp hệ thống tùy biến nhân vật và hệ thống thời trang DIY. Vì vậy, bạn có thể thiết kế quần áo và đồ trang sức của riêng mình bằng cách sử dụng các màu sắc và kết cấu khác nhau mà không gây nhầm lẫn hoặc trùng lặp với những người chơi khác.

3. City of Steam

City of Steam được tuyên bố là một trò chơi trình duyệt có đồ họa hạng nhất, phong cách và giao diện trò chơi tương tự như Diablo 3. Khi tham gia vào Game, người chơi sẽ du hành đến thời Trung cổ - nơi những con quái vật mạnh mẽ và những vị thần giúp con người đánh bại cái ác.

Trong trò chơi này, người chơi sẽ trở thành một trong những anh hùng, sử dụng kiếm, thương và các yếu tố ma thuật để chiến đấu chống lại quái vật với những cỗ máy vô cùng hiện đại nhưng cũng không kém phần thử thách và nguy hiểm. Trò chơi được đón nhận nồng nhiệt nhờ đồ họa, nhưng nó vẫn có một số hạn chế về lối chơi, vì nó phụ thuộc quá nhiều vào Diablo. Tuy nhiên, đây vẫn là một trò chơi đáng chơi dành cho những người hâm mộ đồ họa đỉnh cao của nền tảng trực tuyến

4. Thủy Hử 3D

Đây là Game mobile độc quyền, được phát triển bởi TeamTop3, SohaGame phát hành, phục vụ người dùng iOS và Android. Game được xây dựng dựa trên cốt truyện của bộ tiểu thuyết Thủy hử, tái hiện chân thực những sự kiện, truyền thuyết, điển tích xoay quanh cuộc đời của 108 vị anh hùng Lương Sơn Bạc.

Khi tham gia, người chơi sẽ nhập vai Tử Tinh Đồng Tử - một nhân vật có nhiệm vụ tìm kiếm, thu phục và dẫn dắt các anh hùng Lương Sơn đi đúng đường, giải cứu và chống lại nguy hiểm. Cùng với đồ họa dẫn đầu thị trường lúc bấy giờ, Thủy Hử 3D chính là “kẻ mới nổi” hấp dẫn nhất mùa hè 2013 trong số các Game của Unity tại Việt Nam.

2.6. Ngôn ngữ lập trình trong môi trường Unity – C Sharp (C#)



Hình 2. 8 Ngôn ngữ lập trình C#⁸

2.6.1. C# là gì?

C# là một ngôn ngữ lập trình được phát triển bởi Microsoft vào cuối những năm 1990 và đầu những năm 2000. Đây là một trong những ngôn ngữ lập trình chính của nền tảng .NET Framework của Microsoft.

C# được thiết kế để cung cấp một ngôn ngữ lập trình dễ học, dễ sử dụng và mạnh mẽ cho việc phát triển ứng dụng trên nền tảng Windows. Tuy nhiên, C# cũng

⁸ <https://fptshop.com.vn/tin-tuc/danh-gia/c-la-gi-175870>

đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới và được sử dụng trong nhiều lĩnh vực khác nhau, bao gồm phát triển ứng dụng web, ứng dụng di động, ứng dụng desktop, và đặc biệt là trong việc phát triển trò chơi.

2.6.2. Ưu điểm của C#

- Ngôn ngữ dễ học và sử dụng: C# được thiết kế để có cú pháp gần gũi với ngôn ngữ tự nhiên, giúp người lập trình dễ dàng học và sử dụng.
- Hỗ trợ mạnh mẽ từ Microsoft: C# là một phần của nền tảng .NET của Microsoft, có sự hỗ trợ mạnh mẽ từ công ty, bao gồm các công cụ phát triển và tài liệu hướng dẫn.
- Đa năng và đa nền tảng: C# có thể được sử dụng cho nhiều loại ứng dụng khác nhau, từ ứng dụng web, ứng dụng di động đến ứng dụng desktop và phát triển game. Nó cũng có thể chạy trên nhiều hệ điều hành khác nhau như Windows, Linux và macOS.
- Hỗ trợ lập trình hướng đối tượng (OOP): C# hỗ trợ các tính năng của lập trình hướng đối tượng như kế thừa, đa hình, đóng gói và trừu tượng hóa, giúp tổ chức mã nguồn một cách dễ dàng và hiệu quả.
- Cộng đồng lớn và sôi động: C# có một cộng đồng lập trình viên lớn và sôi động, cung cấp nhiều tài liệu, thư viện mã nguồn mở và hỗ trợ trực tuyến.

2.6.3. Nhược điểm của C#

- Phụ thuộc vào nền tảng .NET: C# được phát triển bởi Microsoft và phụ thuộc vào nền tảng .NET Framework hoặc .NET Core, điều này có nghĩa là ứng dụng C# thường có kích thước lớn hơn và yêu cầu sự hỗ trợ của .NET Framework hoặc .NET Core trên máy tính chạy.
- Hiệu suất: Mặc dù C# có thể tạo ra các ứng dụng hiệu suất cao, nhưng hiệu suất của nó thường không được coi là tốt như các ngôn ngữ khác như C++ trong một số trường hợp đặc biệt như game yêu cầu xử lý nhanh và mượt mà.
- Quản lý bộ nhớ tự động: Trong C#, quản lý bộ nhớ tự động do garbage collector (bộ thu gom rác) thực hiện có thể gây ra các vấn đề hiệu suất trong

các ứng dụng yêu cầu xử lý nhanh và hiệu suất cao.

- Tích hợp không hoàn toàn tốt với các ngôn ngữ khác: Mặc dù C# có thể được tích hợp với các ngôn ngữ khác, nhưng tích hợp này không hoàn toàn tốt như trong một số ngôn ngữ khác như Python hoặc JavaScript.

CHƯƠNG 3.

XÂY DỰNG PHẦN MỀM TRÒ CHƠI 2D ĐA NỀN TẢNG SỬ DỤNG UNITY

3.1. Đặc tả phần mềm

Nền tảng: Windows, Android.

Thể loại game: Platform, Phiêu lưu.

Phần mềm phát triển: Unity Engine.

Đối tượng người chơi hướng tới: Người chơi độ tuổi từ 12 – 35 tuổi

3.1.1. *Game Player*

Nhân vật chính khám phá thế giới mới, vượt qua chướng ngại thử thách hoàn thành nhiệm vụ đánh quái đánh boss, để hoàn thành trò chơi.

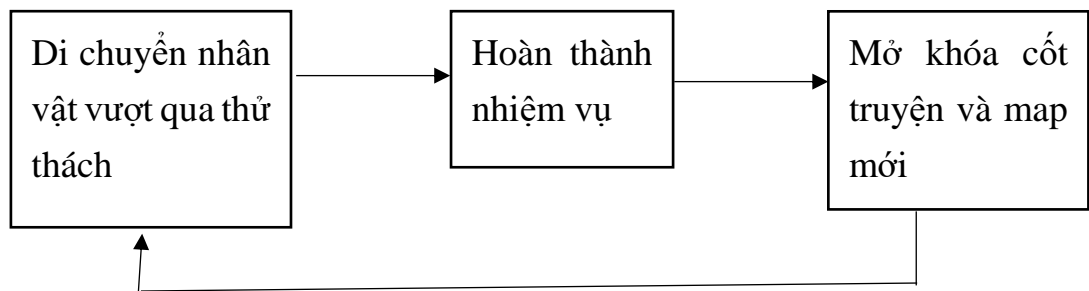
3.1.2. *Player Experience*

Khi chơi game, người chơi sẽ có cảm giác vui vẻ khi qua 1 thử thách. Tức giận khi thất bại khi vượt qua thử thách. Hồi hộp khi chờ đợi thử thách tiếp theo. Nhập vai nhân vật khám phá thế giới.

3.1.3. *Core Mechanism*

Người chơi nhận nhiệm vụ từ npc sử dụng kỹ năng run, jump, dash, hold, jump wall để vượt qua thử thách chướng ngại và hoàn thành nhiệm vụ.

3.1.4. *Core Loop*



3.1.5. *Cốt truyện*

Chương 1:

Nhân vật chính đi lạc và bị hút vào một lỗ đen. Bên kia lỗ đen là một thế giới hoàn toàn mới. Nhân vật chính gặp, làm quen và kết bạn với một người bản địa.

Người bạn đã dẫn nhân vật chính đi khám phá thế giới mới này. Cuối chương người bạn mời nhân vật chính tới làng mình.

Triển khai cốt chuyện: Player chạm vào lỗ đen được chuyển sang màn 1. Player chạm vào npc người bạn và được npc người bạn giao nhiệm vụ thám hiểm vùng đất mới và thu thập đủ số lượng trái cây. Báo cáo nhiệm vụ để kết thúc màn 1 và chuyển sang màn 2

Chương 2.

Nhân vật chính được người bạn dẫn tới ngôi làng của mình. Từ xa người bạn phát hiện ngôi làng đang bị quái vật tấn công. Nhân vật chính gặp trưởng làng và được anh ta mời giúp ngôi làng đánh đuổi quái vật. Nhân vật chính cùng người bạn đã đánh bại tất cả quái vật giải cứu ngôi làng. Sau cuộc chiến trưởng làng đã cảm ơn nhân vật chính và mở tiệc để chiêu đãi cậu.

Triển khai cốt truyện: Player chạm vào npc trưởng làng kích hoạt nhiệm vụ đánh quái. Sau khi đánh hết quái báo cáo nhiệm vụ với npc trưởng làng để hoàn thành màn 2 chuyển sang màn 3.

Chương 3

Sau khi cuộc chiến kết thúc, người bạn phát hiện em gái của mình đã biến mất. Trưởng làng cho rằng em gái đã bị bọn quái vật bắt đi và nhốt tại sào huyệt của chúng. Nhân vật chính và người bạn đã cùng nhau tiến về sào huyệt của bọn quái vật để giải cứu em gái. Trên đường đi giải cứu em gái, nhân vật chính và người bạn gặp rất nhiều thử thách, cạm bẫy, và rất nhiều quái vật tấn công. Sau khi vượt qua hết tất cả cạm bẫy và quái vật, nhân vật chính đã tiếp cận sào huyệt của bọn quái vật. Tuy rất cẩn thận để không bị phát hiện, nhưng nhân vật chính vẫn bị bọn quái vật phát hiện. Nhân vật chính đối mặt với quái vật và thủ lĩnh của chúng. Sau một cuộc chiến gian khổ thủ lĩnh của bọn quái vật bị đánh bại. Các con quái vật còn lại thì nhau chạy trốn. Nhân vật chính gặp được em gái của người bạn và đưa em ấy an toàn về làng.

Triển khai cốt chuyện: Player sử dụng kỹ năng của mình vượt qua cạm bẫy và đánh quái. Đi được đến cuối sẽ gặp được boss, sử dụng kỹ năng để tấn công boss. Sau khi boss chết npc em gái xuất hiện, Player chạm vào npc em gái để hoàn thành trò chơi

3.2. Phân tích thiết kế

3.2.1. Các tác nhân của game

1. Người chơi

a. Điều khiển nhân vật

- Người chơi có thể điều khiển và điều hướng nhân vật chính của trò chơi thông qua các phím điều khiển, joystick hoặc cử chỉ trên màn hình.
- Hoạt động này bao gồm di chuyển nhân vật, nhảy qua chướng ngại vật, tấn công kẻ thù và thực hiện các hành động khác để tiến bộ trong trò chơi.

b. Tương tác với giao diện người dùng (UI)

- Người chơi có thể tương tác với các yếu tố UI như menu, bảng điều khiển, thông báo và nút chức năng để thực hiện các hành động như bắt đầu trò chơi, chọn cấp độ, quản lý cài đặt và xem thông tin.
- Họ cũng có thể sử dụng các yếu tố UI để kiểm tra thông tin như số điểm, mạng sống, tiền xu hoặc điểm kỹ năng.

c. Giao tiếp với môi trường trò chơi

- Người chơi có thể tương tác với môi trường trò chơi bằng cách nhấn vào các đối tượng tương tác như cửa, công tắc hoặc vật phẩm.
- Họ cũng có thể tham gia vào các hoạt động tương tác như thu thập vật phẩm, giải quyết câu đố hoặc tìm kiếm bí mật trong môi trường.

d. Quản lý tài nguyên và kỹ năng

- Người chơi có thể quản lý tài nguyên như máu, mana, vật phẩm và tiền trong trò chơi.
- Họ cũng có thể tăng cường kỹ năng và khả năng của nhân vật bằng cách sử dụng điểm kỹ năng hoặc thu thập và sử dụng vật phẩm.

3.2.2. Các chức năng của game

1. Hệ Thống Lưu Trữ và Tiến Trình:

Chức năng này cho phép người chơi lưu trạng thái hiện tại của trò chơi và tiếp tục từ điểm đã lưu trước đó. Điều này giúp họ có thể tiếp tục trò chơi từ nơi họ đã dừng lại mà không cần phải bắt đầu lại từ đầu.

Dưới đây là cụ thể lưu trữ level màn chơi:

- Người chơi mặc định mở khóa level 1
- Người chơi khi hoàn thành level sẽ được mở khóa level tiếp theo
- Người chơi không thể chơi các level chưa được mở khóa

2. Câu Chuyện và Nhiệm Vụ:

Trò chơi có một câu chuyện hoặc loạt nhiệm vụ mà người chơi phải hoàn thành. Các nhiệm vụ này có thể bao gồm việc giải quyết câu đố, chiến đấu với kẻ thù, hoặc khám phá môi trường để tiến lên trên bản đồ và tiết lộ phần mới của câu chuyện.

Dưới đây là mô tả nhiệm vụ:

- Player nhận nhiệm vụ từ npc
- Npc giao nhiệm vụ
- Nhiệm vụ tìm hoa quả
- Nhiệm vụ tiêu diệt quái vật
- Sau khi hoàn thành nhiệm vụ sẽ được mở khóa map mới



Hình 3. 1 Giao diện Nhiệm vụ thu thập hoa quả



Hình 3. 2 Giao diện nhiệm vụ đánh đuổi quái vật

16. Hệ thống sức khỏe của Player

Hệ thống sức khỏe của người chơi là một yếu tố quan trọng trong trò chơi, điều này quyết định sức mạnh và khả năng sống sót của nhân vật trong cuộc chiến. Dưới đây là một mô tả chi tiết về hệ thống sức khỏe của người chơi:

- Player có max life là cố định
- Khi Life bằng 0 thì game Over
- Khi life lớn hơn max life thì life bằng max life.
- Khi người chơi mất mạng thì life sẽ bị trừ đi
- Khi người chơi được thêm mạng thì life sẽ được cộng vào

3. Hệ Thống Cấp Độ và Tiến Trình:

Người chơi có thể tăng cường kỹ năng và khả năng của nhân vật của họ thông qua việc thu thập điểm kinh nghiệm hoặc hoàn thành nhiệm vụ. Mỗi lần đạt cấp độ mới, họ có thể mở khóa các kỹ năng và tính năng mới.

Dưới đây là chi tiết tiến trình mở khóa kỹ năng

- Khi bắt đầu người chơi chỉ có kỹ năng di chuyển, nhảy, nhảy cao, bám tường, nhảy tường.
- Khi hoàn thành nhiệm vụ hoặc tới màn chơi đặc biệt người chơi sẽ được mở khóa kỹ năng dịch chuyển

4. Hệ Thống Phần Thưởng và Khích Lệ:

Khi người chơi hoàn thành các nhiệm vụ hoặc đạt được mục tiêu cụ thể, họ sẽ nhận được các phần thưởng như vật phẩm, tiền xu, điểm kỹ năng hoặc bảo thưởng. Những phần thưởng này giúp khích lệ người chơi tiếp tục chơi và hoàn thành nhiệm vụ.

Dưới đây là chi tiết chức năng

- Người chơi sau khi hoàn thành màn chơi sẽ được đánh giá từ một đến ba sao tùy vào khả năng hoàn thành nhiệm vụ của nhân vật

5. Hệ thống bộ đếm thời gian

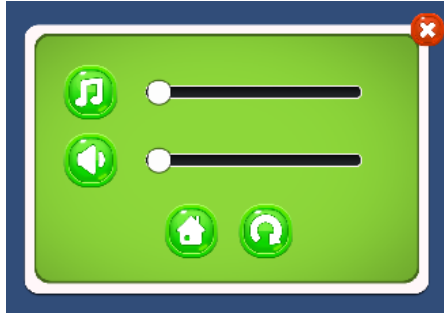
Khi Game được bắt đầu, thì hệ thống bộ đếm thời gian sẽ tiến hành đếm thời gian người chơi đã chơi cho đến khi kết thúc game.

6. Hệ Thống Giao Diện Người Dùng (UI):

Giao diện người dùng cung cấp các phần tử như menu, bảng điều khiển, thông báo và hướng dẫn. Các yếu tố UI này giúp người chơi tương tác và điều chỉnh trò chơi theo ý muốn, bao gồm cài đặt, quản lý nhân vật, và xem thông tin trò chơi.

Dưới đây là chi tiết chức năng

a. Giao diện điều khiển âm lượng



Hình 3. 3 Giao diện Cài đặt âm thanh

- Người chơi có thể tăng hoặc giảm âm lượng nhạc nền hoặc âm thanh của trò chơi
- Người chơi có thể bật hoặc tắt nhạc nền hoặc âm thanh của trò chơi.

b. Giao diện game Start

- Người chơi có thể chọn một trong những màn chơi đã mở khóa để chơi
- Người chơi có thể xem số sao đã đạt được qua từng màn
- Người chơi có thể chọn xem hướng dẫn của trò chơi
- Người chơi có thể nhấn chơi màn chơi cao nhất đã mở khóa
- Người chơi có thể nhấn thoát trò chơi

c. Giao diện game Play



Hình 3. 4 Giao diện Game Play

- Người chơi có thể nhìn thấy số máu của Player
- Người chơi có thể nhìn thấy nhiệm vụ phải thực hiện và tiến độ thực hiện

d. Giao diện game Over



Hình 3. 5 Giao diện Game Over

- Người chơi sẽ nhìn thấy tiến độ thực hiện nhiệm vụ sau khi game over
- Người chơi có thể nhấn nút chơi lại màn chơi
- Người chơi có thể nhấn nút trở lại màn hình Start

e. Giao diện game Complete



Hình 3. 6 Giao diện game Complete

- Người chơi có thể nhìn thấy số sao đạt được
- Người chơi sẽ nhìn thấy tiến độ thực hiện nhiệm vụ sau khi game complete
- Người chơi có thể nhấn nút chơi lại màn chơi
- Người chơi có thể nhấn nút trở lại màn hình Start
- Người chơi có thể nhấn nút để chuyển sang màn tiếp theo

6. Hệ Thống Âm Thanh và Hiệu Ứng Đặc Sắc:

Âm thanh và hiệu ứng đặc sắc giúp tạo ra một trải nghiệm trò chơi sống động và hấp dẫn hơn. Điều này bao gồm nhạc nền, hiệu ứng âm thanh cho các hành động trong trò chơi, và âm thanh môi trường.

Dưới đây là chi tiết chức năng

- Mỗi màn chơi sẽ có nhạc nền khác nhau
- Tương tác giữa player và môi trường sẽ tạo ra hiệu ứng âm thanh và hình ảnh

7. Hệ Thống Cài Đặt và Tùy Chỉnh:

Hệ thống này cho phép người chơi điều chỉnh cài đặt như độ khó, ngôn ngữ, âm thanh, và các tùy chọn tùy chỉnh khác, để tạo ra một trải nghiệm chơi game cá nhân hóa theo sở thích của mình.

8. Hệ thống đối thoại với npc

Hệ thống này cho phép người chơi tương tác với npc. Đối thoại với các npc qua đó hiểu được cốt truyện của game



Hình 3. 7 Giao diện Hội thoại

3.2.3. Các đối tượng game

Các đối tượng game gồm Player, Spike, Saw, Spike Head, Rock Head, Spike Ball, Fire, Platform, Box, Trampoline, Furit, Heart, Check point.

1. Player

Là nhân vật chính của game. Người chơi sẽ điều khiển player để vượt qua màn chơi



a. Move

- Nhân vật có thể di chuyển sang trái hoặc phải
- Dùng gia tốc để nhân vật di chuyển được chân thật

b. Jump

- Nhân vật sẽ nhảy lên một đoạn
- Nhân vật sẽ nhảy cao hơn khi nhấn giữ
- Khi nhân vật chạm đất mới nhảy được
- Sử dụng cơ chế coyote time và jump buffering để nhân vật nhảy được mượt hơn

c. Dash

- Nhân vật dịch chuyển một đoạn theo hướng chỉ định
- Hướng chỉ định là 8 hướng được tạo từ 4 phím mũi tên
- Nhân vật chỉ được dash 1 lần khi ở trên không, muốn dash tiếp thì phải chạm đất
- Khi Dash thì nhân vật không thể hành động nào khác
- Khi Dash vào tường thì sẽ dừng lại ngay
- Khi chạm tường thì không dash được
- Sử dụng cơ chế coyote time để nhân vật được Dash mượt hơn.

d. Hold

- Nhân vật bám vào tường
- Khi chạm tường mới có thể Hold
- Bám tường trong thời gian cố định, quá thời gian sẽ bị hết thể lực
- Nhân vật chạm đất sẽ được hồi thể lực

e. Jump Wall

- Nhân vật nhảy ra khỏi bức tường
- Khi chạm tường và không chạm đất mới nhảy được

f. Die

- Nhân vật sẽ mất mạng

- Sau khi mất mạng nhân vật sẽ ở trạng thái bất khả chiến bại trong 0.3s

g. Healing

- Nhân vật sẽ được hồi mạng sống

2. Người bạn



- Player sẽ trò chuyện với npc người bạn và nhận nhiệm vụ “Tìm kiếm thức ăn”

3. Trưởng làng



- Player sẽ trò chuyện với npc trưởng làng và nhận nhiệm vụ “Đánh đuổi quái vật”

4. Em gái người bạn



- Player sẽ trò chuyện với npc em gái và hoàn thành trò chơi.

5. Spike



- Player va chạm với Spike sẽ mất một mạng.

6. Saw



- Saw di chuyển tự động tuần tự qua các điểm.
- Saw ở trạng thái đứng yên thì có thể kích hoạt cho Saw di chuyển
- Saw có thể di chuyển liên tục hoặc không
- Saw có thể di chuyển có gia tốc hoặc không
- Player va chạm với Saw sẽ mất một mạng.

7. Spike Head



- Spike head di chuyển tự động tuần tự qua các điểm
- Spike head ở trạng thái đứng yên thì có thể kích hoạt cho Spike head di chuyển
- Spike head có thể di chuyển liên tục hoặc không
- Spike head có thể di chuyển có gia tốc hoặc không
- Player va chạm với Spike head sẽ mất một mạng.

8. Rock Head



- Rock head di chuyển tự động tuần tự qua các điểm
- Rock head ở trạng thái đứng yên thì có thể kích hoạt cho Rock head di chuyển
- Rock head có thể di chuyển liên tục hoặc không
- Rock head có thể di chuyển có gia tốc hoặc không
- Rock head trong lúc di chuyển va chạm với Player sẽ mất một mạng

9. Spike Ball



- Trạng thái 1: Spike Ball lắc qua lắc lại theo trục với tốc độ chậm dần hai bên và nhanh dần ở giữa.
- Trạng thái 2: Spike Ball xoay tròn quanh trục với tốc độ không đổi.
- Player va chạm với Spike Ball sẽ mất một mạng.

10. Fire



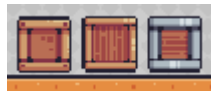
- Khi Player chạm vào Fire sau 0.3s Fire được kích hoạt
- Fire khi được kích hoạt sẽ tạo ra một ngọn lửa. Nếu player chạm vào ngọn lửa sẽ mất 1 mạng
- Fire sau khi được kích hoạt sẽ tắt sau 3s.

11. Platform



- Platform di chuyển tự động tuần tự qua các điểm.
- Platform ở trạng thái đứng yên thì có thể kích hoạt cho Platform di chuyển bằng cách đứng trên Platform.
- Platform có thể di chuyển liên tục hoặc không
- Platform có thể di chuyển có gia tốc hoặc không

12. Box



- Khi player nhảy lên trên box. Box sẽ bị mất máu.
- Khi máu của Box về 0 nó sẽ vỡ ra các mảnh vỡ.
- Sau khi hộp vỡ ra có thể xuất hiện các vật phẩm.
- Khi hộp vỡ player sẽ bị phản lại 1 lực hướng lên.

13. Trampoline



- Khi nhân vật nhảy lên Trampoline thì nhân vật sẽ bị bật lên một đoạn

14. Furit



- Là vật phẩm nhiệm vụ thu thập để hoàn thành nhiệm vụ
- Sẽ xuất hiện lại sau 60s sau khi bị kích hoạt

15. Heart



- Vật phẩm hồi máu cho player
- Player khi va chạm với Heart thì sẽ thêm mạng
- Khi Player đầy mạng thì sẽ được cộng thêm 1 mạng vào hp gốc

16. Check point



- Điểm lưu lại vị trí cho phayer sau khi mất mạng sẽ trở lại
- Check point được kích hoạt thì lưu lại vị trí này của Check point vào điểm reset.
- Nhân vật sau khi chết sẽ quay lại vị trí này

17. Bee



- Là một Enemy tấn công trên không
- Bee không có khả năng di chuyển
- Khi Player vào tầm bắn của bee. Bee sẽ tự động tìm đến player và bắn ra gai nhọn để tấn công

18. Plant



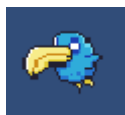
- Là một Enemy tấn công dưới đất
- Plant không có khả năng di chuyển
- Khi player vào tầm bắn của Plant, Plant sẽ bắn ra hạt để tấn công

19. Mushroom



- Là một Enemy không có khả năng tấn công và di chuyển trên mặt đất
- Mushroom di chuyển qua lại giữa 2 điểm

20. Blue Bird



- Là một Enemy không có khả năng tấn công và bay trên không
- Blue Bird di chuyển qua lại giữa 2 điểm

21. Chicken



- Là một Enemy không có khả năng tấn công và di chuyển trên mặt đất
- Chicken di chuyển qua lại giữa 2 điểm

22. Chameleon



- Là một enemy tấn công và di chuyển trên mặt đất
- Chameleon di chuyển qua lại giữa 2 điểm
- Khi player đi vào tầm phát hiện của Chameleon, Chameleon sẽ di chuyển tới player. Sau khi di chuyển tới một khoảng cách nhất định thì tấn công

23. Radish



- Là một enemy tấn công và di chuyển trên mặt đất
- Radish di chuyển qua lại giữa 2 điểm
- Khi player đi vào tầm phát hiện của Radish, Radish sẽ di chuyển tới player. Sau khi di chuyển tới một khoảng cách nhất định thì bắn ra lá cây trên đầu để tấn công

24. Rino



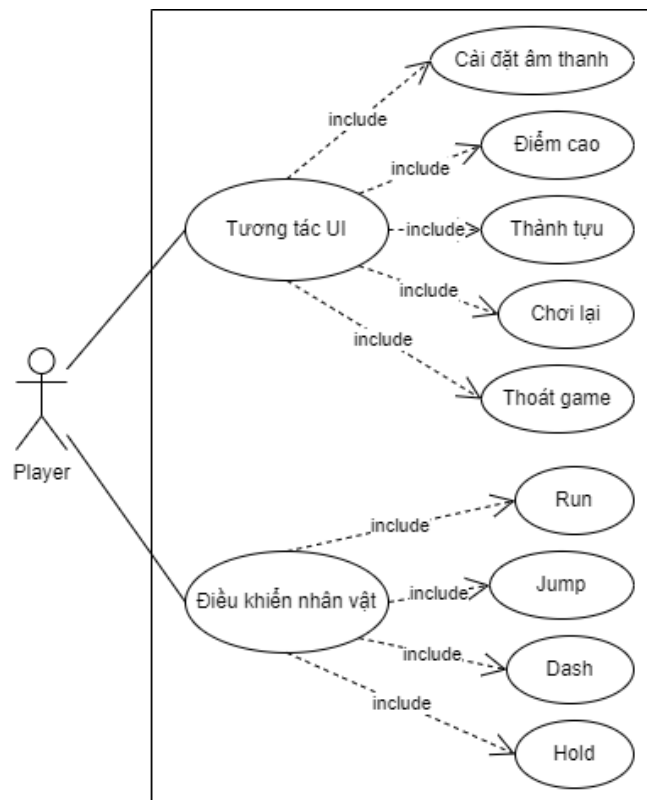
- Là một enemy tấn công và di chuyển trên mặt đất
- Rino di chuyển qua lại giữa 2 điểm
- Trên đường di chuyển Rino sẽ va vào player để tấn công

25. Trunk

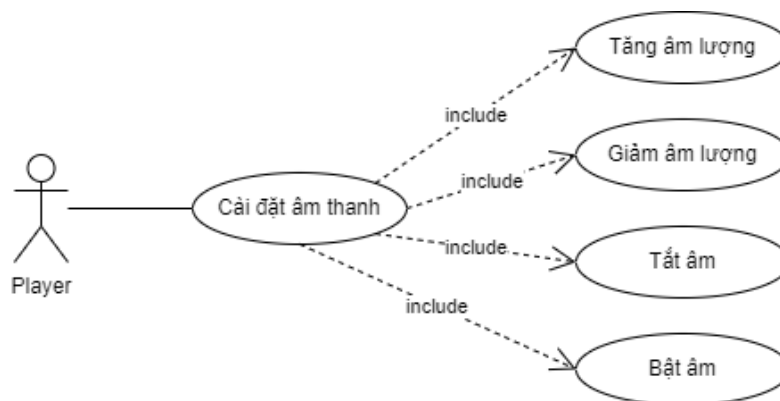


- Là một enemy tấn công và di chuyển trên mặt đất
- Trunk di chuyển qua lại giữa 2 điểm
- Khi player đi vào tầm phát hiện của Trunk, Trunk sẽ di chuyển tới player. Sau khi di chuyển tới một khoảng cách nhất định thì bắn ra khúc gỗ để tấn công

3.2.4. Biểu đồ Use – case



Hình 3. 8 Biểu đồ Use case tổng quát



Hình 3. 9 Biểu đồ thành phần Chức năng cài đặt âm thanh

1. Đặc tả use case chức năng cài đặt âm thanh

Mã use case	UC01		
Tên use case	Use case cài đặt âm thanh		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng chỉnh sửa âm thanh		
Luồng sự kiện		Tác nhân	Hệ thống phản hồi
	1	Chọn biểu tượng cài đặt	
	2		Đưa ra giao diện cài đặt
	3	Chỉnh sửa âm thanh	
	4		Hệ thống thực hiện điều chỉnh âm thanh như cài đặt
Điều kiện sau	Hệ thống lưu dữ liệu thay đổi vào máy		
Điều kiện thoát	Người dùng lựa chọn thoát		

Bảng 3. 1 Đặc tả use case chức năng cài đặt âm thanh

2. Đặc tả use case Xem điểm cao

Mã use case	UC02		
Tên use case	Use case xem điểm cao nhất		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng xem điểm cao nhất đạt được		
Luồng sự kiện		Tác nhân	Hệ thống phản hồi
	1	Mở giao diện Game Start	
	2		Đưa ra giao diện Game Start
	3	Chọn nút Điểm cao	
	4		Hiển thị điểm cao ra màn hình

Mã use case	UC02
Điều kiện thoát	Người dùng lựa chọn thoát

Bảng 3. 2 Đặc tả use case Xem điểm cao

3. Đặc tả use case xem thành tựu

Mã use case	UC03		
Tên use case	Use case xem thành tựu		
Tác nhân	Người dùng		
Mô tả	Cho phép người Xem thành tựu đạt được trong game		
Luồng sự kiện		Tác nhân	Hệ thống phản hồi
	1	Mở giao diện Game Start	
	2		Đưa ra giao diện Game Start
	3	Chọn nút Thành tựu	
	4		Hiển thị thành tựu ra màn hình
Điều kiện thoát	Người dùng lựa chọn thoát		

Bảng 3. 3 Đặc tả use case xem thành tựu

4. Đặc tả use case chức năng chơi lại

Mã use case	UC04		
Tên use case	Use case chơi lại		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng chơi lại màn chơi		
Luồng sự kiện		Tác nhân	Hệ thống phản hồi
	1	Chọn biểu tượng cài đặt	
	2		Đưa ra giao diện cài đặt
	3	Chọn nút chơi lại	

Mã use case	UC04		
	4		Load lại màn chơi đang chạy
Điều kiện thoát	Người dùng lựa chọn thoát		

Bảng 3. 4 Đặc tả use case chức năng chơi lại

5. Đặc tả chức năng Thoát trò chơi

Mã use case	UC05		
Tên use case	Use case Thoát trò chơi		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng thoát khỏi trò chơi		
Luồng sự kiện		Tác nhân	Hệ thống phản hồi
	1	Mở màn hình Game Start	
	2		Đưa ra giao diện Game Start
	3	Chọn nút Thoát	
	4		Thực hiện đóng trò chơi
Điều kiện thoát	Người dùng lựa chọn thoát		

Bảng 3. 5 Đặc tả chức năng Thoát trò chơi

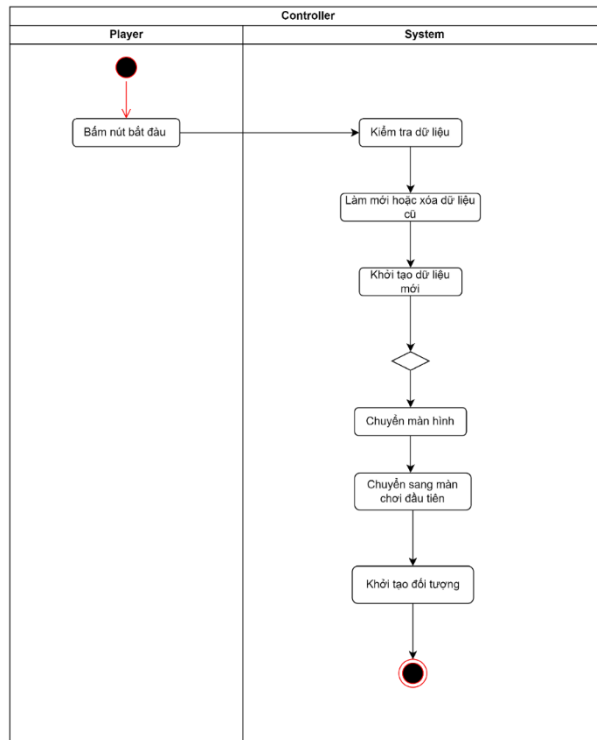
6. Đặc tả use case điều khiển nhân vật

Mã use case	UC06		
Tên use case	Use case Di chuyển nhân vật		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng điều khiển nhân vật game		
Luồng sự kiện		Tác nhân	Hệ thống phản hồi
	1	Chọn nút điều khiển	
	2		Nhân vật di chuyển theo lệnh của nút điều khiển

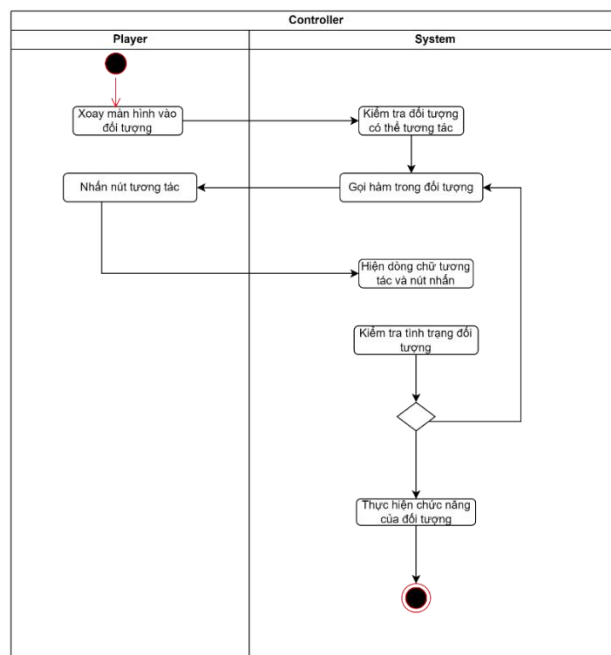
Mã use case	UC06
Điều kiện thoát	Người dùng lựa chọn thoát

Bảng 3. 6 Đặc tả use case điều khiển nhân vật

3.2.4. Biểu đồ hoạt động (Activity Diagram)

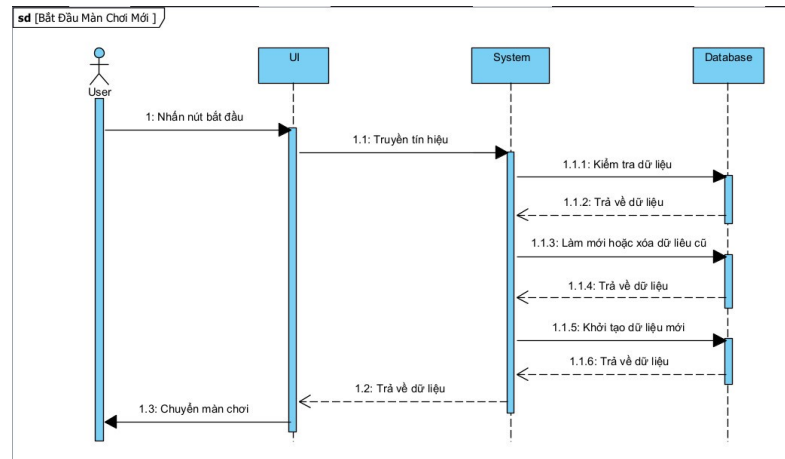


Hình 3. 10 Biểu đồ hoạt động bắt đầu game

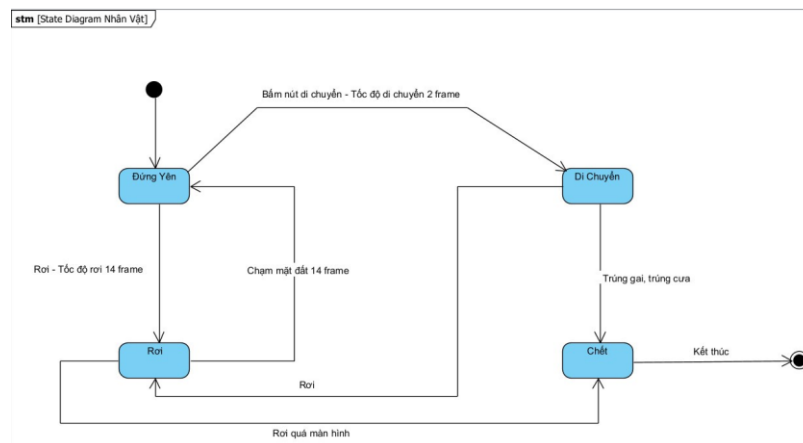


Hình 3. 11 Biểu đồ hoạt động điều khiển nhân vật

3.2.5. Biểu đồ tuần tự (Sequence diagram)

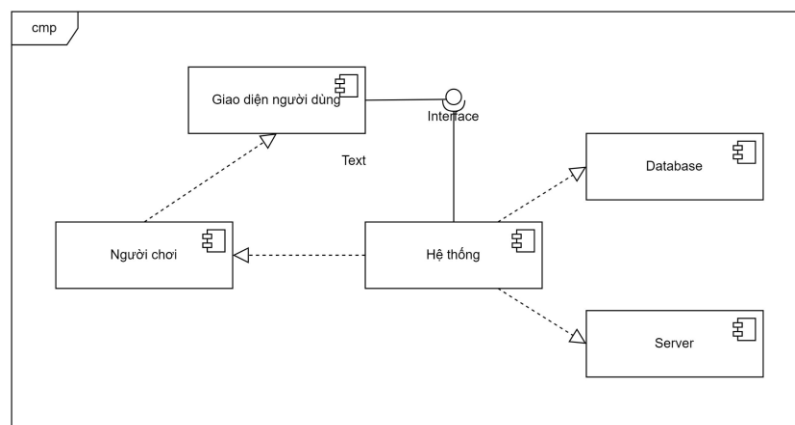


Hình 3. 12 Biểu đồ tuần tự bắt đầu game



Hình 3. 13 Biểu đồ tuần tự điều khiển nhân vật

3.2.6. Biểu đồ thành phần (Component Diagram)



Hình 3. 14 Biểu đồ thành phần

3.3. Các bước cài đặt game

3.3.1. Thiết lập môi trường phát triển

- **Tải và cài đặt Unity Engine:** Truy cập trang chủ của Unity để tải và cài đặt phiên bản Unity phù hợp với hệ điều hành của bạn.
- **Tạo một dự án game 2D mới trong Unity:** Khởi chạy Unity, chọn "New Project", chọn template "2D", đặt tên dự án và chọn vị trí lưu trữ dự án.

3.3.2. Thiết kế game

- **Xác định ý tưởng và cốt truyện của game:** Phác thảo ý tưởng chính của game, bao gồm cốt truyện, mục tiêu và cách chơi.
- **Thiết kế các đối tượng, nhân vật, cảnh quan và giao diện người dùng:** Sử dụng các công cụ như Photoshop, Illustrator để tạo hình ảnh cho các đối tượng trong game.
- **Tạo tài nguyên như hình ảnh, âm thanh, hiệu ứng:** Chuẩn bị các tài nguyên hình ảnh (sprites), âm thanh (sound effects, background music) và các hiệu ứng đặc biệt (particle effects).

3.3.3. Xây dựng cơ bản

- **Tạo các đối tượng game (GameObjects) trong scene:** Sử dụng Unity Editor để thêm các GameObjects vào scene.
- **Thiết lập các thành phần (components) như Sprites, Colliders, Rigidbodies:** Thêm các component cần thiết vào GameObjects để xác định hành vi và tính chất của chúng.
- **Viết script để xử lý logic game, di chuyển và tương tác:** Sử dụng C# để viết các script điều khiển logic của game, như di chuyển nhân vật, xử lý sự kiện và tương tác giữa các đối tượng.

3.3.4. Thiết lập vật lý và va chạm

- **Định nghĩa các lực tác dụng lên các đối tượng như trọng lực, ma sát:** Sử dụng Rigidbodies và các thuộc tính vật lý trong Unity để mô phỏng lực tác động.
- **Cài đặt hệ thống va chạm, xử lý các va chạm giữa các đối tượng:** Sử dụng Colliders và viết script để xử lý va chạm và các hành động xảy ra khi va chạm.

3.3.5. *Điều khiển nhân vật và camera*

- **Tạo script để điều khiển nhân vật, bao gồm di chuyển, nhảy, tấn công:** Viết các script điều khiển nhân vật chính, cho phép nhân vật thực hiện các hành động như di chuyển, nhảy và tấn công.
- **Thiết lập camera để theo dõi và hiển thị nhân vật:** Sử dụng các script và component như Cinemachine để thiết lập camera theo dõi nhân vật và hiển thị các khung cảnh trong game.

3.3.6. *Thêm âm thanh và hiệu ứng*

- **Nhập các tập tin âm thanh như nhạc nền, hiệu ứng âm thanh:** Thêm các file âm thanh vào dự án Unity và cấu hình chúng trong AudioSource.
- **Cài đặt âm thanh và hiệu ứng trong game:** Sử dụng AudioSource để phát nhạc nền, hiệu ứng âm thanh và đồng bộ với các sự kiện trong game.

3.3.7. *Xây dựng giao diện người dùng*

- **Thiết kế và tạo các phần tử giao diện như menu, điểm số, thanh máu:** Sử dụng Unity UI để thiết kế và bố trí các phần tử giao diện người dùng.
- **Viết script để tương tác với các phần tử UI:** Sử dụng các script để xử lý tương tác người dùng với các phần tử UI như nút bấm, thanh trượt, và các chỉ số hiển thị.

3.3.8. *Kiểm thử và điều chỉnh*

- **Thử nghiệm game, phát hiện và sửa chữa các lỗi:** Chạy thử game để phát hiện lỗi, ghi nhận phản hồi và tiến hành sửa lỗi.
- **Tối ưu hóa hiệu suất, cải thiện trải nghiệm người dùng:** Điều chỉnh các thông số để đảm bảo game chạy mượt mà trên nhiều thiết bị, cải thiện giao diện và tương tác người dùng.

3.3.9. *Đóng gói và phát hành*

- **Cấu hình các thiết lập để build game:** Chọn các thiết lập build trong Unity phù hợp với nền tảng mục tiêu.
- **Xuất game cho các nền tảng mong muốn như Windows, Android, iOS:** Sử dụng Unity Build Settings để xuất bản game cho các nền tảng đích, kiểm tra lại để đảm bảo game hoạt động tốt trên các thiết bị mục tiêu.

3.4. Kiểm thử

Step	Test Steps	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Nhận nhiệm vụ từ npc	Người dùng sẽ có thể nhận nhiệm vụ	Người dùng nhận được nhiệm vụ từ npc	Pass	
2	Hệ thống sức khỏe của Player	Player sẽ mất máu, hồi máu và thêm máu	Như mong đợi	Pass	
3	Hệ thống phần thưởng và khích lệ	Player sẽ mở khóa map mới khi hoàn thành nhiệm vụ	Như mong đợi	Pass	
4	Hệ thống điều khiển âm lượng game	Người dùng sẽ có thể điều khiển âm thanh của game	Như mong đợi	Pass	
5	Hệ thống đối thoại với npc	Player có thể tương tác với npc. Đưa ra các cuộc đối thoại	Như mong đợi	Pass	
6	Di chuyển nhân vật	1. Nhấn mũi tên sang phải/Nhấn phím D	Nhân vật quay mặt sang phải	Pass	
		2. Nhấn và giữ mũi tên sang trái/Nhấn giữ phím A	Nhân vật quay mặt sang trái và di chuyển sang trái	Pass	

Step	Test Steps	Expected Result	Actual Result	Status (Pass/Fail)	Notes
		3. Nhấn và giữ mũi tên sang phải/ Nhấn giữ phím D	Nhân vật foggy quay mặt sang phải và di chuyển sang phải	Pass	
		4. Nhấn phím C	Nhân vật nhảy lên	Pass	
		5. Nhấn và giữ C	Nhân vật nhảy cao hơn	Pass	
		6. Nhấn phím X	Nhân vật dịch chuyển 1 đoạn	Pass	
		7. Nhấn phím X và điều khiển hướng bằng các phím mũi tên	Nhân vật dịch chuyển theo hướng chỉ định	Pass	
		8. Nhấn và giữ phím Z khi chạm tường ở trên không	Nhân vật bị dính vào tường	Pass	
		9. Nhấn và giữ phím Z trong khi nhân vật đang bám tường nhấn phím C	Nhân vật bật ra khỏi tường một đoạn	Pass	
7	Thu thập hoa quả	1.Điều khiển nhân chạm vào hoa quả	Hoa quả lập tức biến mất	Pass	

Step	Test Steps	Expected Result	Actual Result	Status (Pass/Fail)	Notes
		2. Nhận nhiệm vụ từ NPC. Điều khiển nhân vật chạm vào hoa quả trong danh sách nhiệm vụ	Hoa quả lập tức biến mất. Hoa quả được thêm vào danh sách nhiệm vụ	Pass	
8	Đánh bại quái vật	1. Điều khiển nhân vật nhảy lên đầu quái vật	Nhân vật bị nhận sát thương và chết	Pass	
		2. Nhận nhiệm vụ từ NPC. Điều khiển nhân vật nhảy lên đầu quái vật theo thứ tự trong danh sách	Nhân vật bị nhận sát thương và chết. Sau khi quái vật chết danh sách nhiệm vụ đã ghi nhận	Pass	
9	Nhân vật bị mất máu	Điều khiển nhân vật va chạm với cạm bẫy	Nhân vật lập tức mất một mạng và dịch chuyển về điểm check point	Pass	

Bảng 3. 7 Test Case

CHƯƠNG 4.

KẾT QUẢ NGHIÊN CỨU VÀ ĐÁNH GIÁ

4.1. Kết quả đạt được

4.1.1. Cơ sở lý thuyết

Trong quá trình thực hiện đồ án, em đã đạt được những kết quả sau:

- **Củng cố kiến thức lý thuyết:** Em đã củng cố và vận dụng những kiến thức lý thuyết đã học để phân tích và giải quyết các vấn đề trong đồ án của mình.
- **Áp dụng kiến thức vào thực tiễn:** Kiến thức lý thuyết đã giúp em áp dụng vào thực tế để xây dựng và triển khai dự án một cách hiệu quả.
- **Rèn luyện kỹ năng quản lý dự án:** Em đã học được kỹ năng quản lý dự án, bao gồm sử dụng Gitlab và thiết lập quy trình UniTest.
- **Phân bố bộ cục và sử dụng thư mục:** Em đã hiểu cách phân bố bộ cục thư mục và tổ chức các tệp tin một cách hợp lý nhất cho dự án.

4.1.2. Cơ sở thực hành

Trong giai đoạn thực hành, em đã đạt được các kết quả sau:

- **Fix bug và bố cục layout:** Em đã tự học cách sửa lỗi và cải thiện bố cục layout cho ứng dụng của mình.
- **Kỹ năng làm việc nhóm:** Mặc dù làm một mình, nhưng em đã rèn luyện kỹ năng làm việc độc lập và tự giải quyết vấn đề.
- **Lập kế hoạch và viết báo cáo:** Em đã tự xây dựng kế hoạch và viết báo cáo cho đồ án của mình.

4.1.3. Kỹ năng đạt được

Cuối cùng, qua quá trình này, em đã rèn luyện và đạt được các kỹ năng sau:

- **Rèn luyện kỹ năng tổ chức công việc:** Em đã học cách tổ chức công việc và quản lý thời gian một cách hiệu quả để đảm bảo tiến độ của dự án.
- **Nắm bắt nội quy và tác phong của dự án:** Em đã hiểu và áp dụng nội quy và tác phong của dự án vào quá trình làm việc của mình.
- **Truyền đạt ý tưởng và quan điểm:** Em đã học cách truyền đạt ý tưởng và quan điểm của mình một cách rõ ràng và chuyên nghiệp.

4.1.4. Kết quả ứng dụng

Link Cài đặt phần mềm:

<https://drive.google.com/drive/folders/1VguFjm8l5ovnACXHy5AGipbJAm4A791D>

Application link demo: <https://youtu.be/6D8byQ2l50s>

4.2. Đánh giá

4.2.1. Ưu điểm

- Dự án đã phản ánh chính xác và vận dụng tốt các kiến thức lý thuyết đã học.
- Quá trình thực hành được thực hiện một cách độc lập và tự chủ.

4.2.2. Nhược điểm

- Mặc dù làm một mình, nhưng áp lực và deadline vẫn gây ra một số khó khăn trong quá trình thực hiện dự án.
- Đề xuất có thể cải thiện cách phân bổ thời gian và quản lý công việc để tăng hiệu suất làm việc.

4.2.3. Đánh giá chung

Dự án đã đạt được một số kết quả quan trọng trong việc áp dụng kiến thức lý thuyết vào thực tiễn và phát triển kỹ năng làm việc độc lập. Tuy nhiên, vẫn còn cơ hội để cải thiện và phát triển thêm các kỹ năng và năng lực trong tương lai.

4.3. Công việc tương lai

Trong tương lai em sẽ bổ sung thêm một số chức năng cho game của mình như chức năng đặt tên cho nhân vật, chức năng tính thời gian hoàn thành game, chức năng điểm cao và thành tích, chức năng lưu game. Tiếp đó em sẽ đưa game của mình ra thị trường để mọi người đều có thể chơi.

KẾT LUẬN

Trong quá trình thực hiện đồ án "Phát Triển Ứng Dụng Trò Chơi 2D Đa Nền Tảng Sử Dụng Unity", em đã trải qua một hành trình học tập và phát triển đầy thách thức và hứa hẹn. Đây không chỉ là một dự án đơn thuần, mà còn là một cơ hội để rèn luyện kỹ năng, khám phá sự sáng tạo và xây dựng một sản phẩm có giá trị thực tế.

Trong quá trình này, em đã học được rất nhiều từ việc nghiên cứu, thiết kế, phát triển và triển khai ứng dụng trò chơi 2D trên nhiều nền tảng khác nhau bằng công cụ Unity. Em đã áp dụng những kiến thức lý thuyết đã học vào thực tiễn, từ việc xây dựng cấu trúc dữ liệu, quản lý tài nguyên, đến việc lập trình gameplay và điều khiển người chơi.

Bên cạnh đó, qua quá trình phát triển ứng dụng, em đã rèn luyện và phát triển nhiều kỹ năng mềm quan trọng như kỹ năng làm việc độc lập, quản lý thời gian, giải quyết vấn đề và giao tiếp.

Dù có những thách thức và khó khăn, nhưng niềm đam mê và sự kiên trì đã giúp em vượt qua mọi trở ngại. Em tự hào với những gì đã đạt được và tin rằng đây là bước đi quan trọng trong sự nghiệp phát triển phần mềm của mình.

Cuối cùng, em xin gửi lời cảm ơn chân thành đến gia đình, bạn bè và các giáo viên đã luôn ủng hộ và động viên em trong suốt thời gian thực hiện đồ án này. Sự hỗ trợ và động viên từ mọi người đã giúp em vượt qua mọi khó khăn và hoàn thành dự án một cách thành công.

Em hy vọng rằng ứng dụng trò chơi 2D này sẽ mang lại niềm vui và trải nghiệm tuyệt vời cho người chơi, và đồng thời là một bước khởi đầu tích cực cho sự phát triển và học hỏi của em trong tương lai.

TÀI LIỆU THAM KHẢO

Danh mục các tài liệu tham khảo:

- [1]. Jesse Schell, Dịch giả: Hùng Vũ; “*Nghệ Thuật Thiết Kế Game*”, 2023, NXB Dân Trí
- [2]. Mike Geig, “*Unity 2018 Game Development in 24 Hours*”, Sams Teach Yourself, 2023
- [3]. Joe Hocking, “*Unity in Action, Third Edition: Multiplatform game development in C#*”, 2020.

Website tham khảo:

- [4]. <https://viblo.asia/p/awake-start-va-mot-so-ham-co-ban-trong-monobehaviour-qPoL77nNLvk>
- [5]. <https://viblo.asia/p/co-ban-ve-engine-vat-ly-trong-unity-3OEqGj1PM9bL>
- [6]. <https://viblo.asia/p/gioi-thieu-ve-unity-engine-game-engine-pho-bien-nhat-hien-nay-V3m5WBj8lO7>
- [7]. <https://fonesmart.com.vn/tin-cong-nghe/he-dieu-hanh-android-la-gi-tim-hieu-chi-tiet-ve-he-dieu-hanh-android-va-nhung-thuong-hieu-dien-thoai-chay-android-noi-tieng-n779.html>
- [8]. <https://www.dienmayxanh.com/kinh-nghiem-hay/he-dieu-hanh-windows-la-gi-596407>
- [9]. <https://bizfly.vn/techblog/webgl-la-gi.html>
- [10]. <https://fptshop.com.vn/tin-tuc/danh-gia/c-la-gi-175870>