**BỘ GIÁO DỤC & ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC PHENIKAA**

**KHOA CÔNG NGHỆ THÔNG TIN**

**-------- \*\*\* --------**



BÁO CÁO DỰ ÁN HỌC PHẦN

PHÂN TÍCH VÀ THIẾT KẾ PHẦN MỀM

Đề tài: Lập trình game Adventure Rabbit 2D

|  |  |
| --- | --- |
| Thành viên: | Đỗ Hoàng Giang – 21012054 |
| Giảng viên hướng dẫn: | Trịnh Thanh Bình |

Hà Nội, Ngày 30 tháng 10 năm 2023

**MỤC LỤC**

[LỜI MỞ ĐẦU 3](#_Toc149416438)

[1. Yêu cầu (Requirements) 5](#_Toc149416439)

[**1.1 Đặt vấn đề (Problem statement)** 5](#_Toc149416440)

[1.1.1 Mô tả vấn đề 5](#_Toc149416441)

[1.1.2 Bối cảnh xu hướng công nghệ hiện nay 6](#_Toc149416442)

[1.1.3 Các giải pháp đã có 8](#_Toc149416443)

[1.1.4 Các giải pháp đề xuất 8](#_Toc149416444)

[1.1.5 Cốt truyện của Game 9](#_Toc149416445)

[2. Thiết kế và triển khai 10](#_Toc149416446)

[**2.1 Các yêu cầu chức năng** 10](#_Toc149416447)

[**2.2 Các yêu cầu phi chức năng** 11](#_Toc149416448)

[**2.3 Các ràng buộc** 11](#_Toc149416449)

[2.3.1 Các ràng buộc về triển khai 11](#_Toc149416450)

[2.3.2 Các ràng buộc về kinh tế 12](#_Toc149416451)

[**2.4. Mô hình hệ thống / Thiết kế giải pháp** 12](#_Toc149416452)

[2.4.1. Các kịch bản của hệ thống 12](#_Toc149416453)

[2.4.2 Mô hình Use-case (Use-case) 14](#_Toc149416454)

[2.4.3 Mô hình lớp và đối tượng 14](#_Toc149416455)

[2.4.4 Các biểu đồ tuần tự 16](#_Toc149416456)

[2.4.5 Các màn hình giao diện người dùng: 16](#_Toc149416457)

[2.4.6 Nhân vật và thiết kế lối chơi 24](#_Toc149416458)

[3. Ngôn ngữ và công nghệ sử dụng 26](#_Toc149416459)

[**3.1 Giới thiệu về ngôn ngữ C#** 26](#_Toc149416460)

[**3.2 Công cụ sử dụng** 28](#_Toc149416461)

[3.2.1 Unity 2023.1.0f1 28](#_Toc149416462)

[3.2.2 Ưu điểm và nhược điểm 29](#_Toc149416463)

[4 Tìm hiểu về unity 30](#_Toc149416464)

[**4.1 Unity là gì?** 30](#_Toc149416465)

[**4.3 Một số thống kê về Unity** 32](#_Toc149416466)

[**4.4 Ưu điểm của Unity** 34](#_Toc149416467)

[5. Tìm hiểu về unity engine 37](#_Toc149416468)

[**5.1 Các thành phần trong Unity Editor** 37](#_Toc149416469)

[**5.2 Các khái niệm cơ bản trong Unity** 39](#_Toc149416470)

[6. Một số thành phần khác của dự án 41](#_Toc149416471)

[**6.1 Kế hoạch dự án** 41](#_Toc149416472)

[**6.2 Các vấn đề đạo đức và làm việc chuyên nghiệp** 43](#_Toc149416473)

[**6.3 Kế hoạch cho kiến thức mới và chiến lược học tập** 44](#_Toc149416474)

[7. Chương trình Demo 45](#_Toc149416475)

[**7.1. Gameplay** 45](#_Toc149416476)

[**7.2. Các chức năng cơ bản** 46](#_Toc149416477)

[8. Kết luận 46](#_Toc149416478)

[**8.1 Kết quả** 46](#_Toc149416479)

[**8.2 Đánh giá** 48](#_Toc149416480)

[8.2.1 Những điểm làm được 48](#_Toc149416481)

[8.2.2 Những điểm chưa làm được 48](#_Toc149416482)

[**8.3 Khó khăn và Học hỏi** 49](#_Toc149416483)

[**8.4 Hướng phát triển** 49](#_Toc149416484)

[**8.5 Tổng kết** 50](#_Toc149416485)

[LỜI CẢM ƠN 51](#_Toc149416486)

[TÀI LIỆU THAM KHẢO 51](#_Toc149416487)

# LỜI MỞ ĐẦU

Trong thời đại công nghệ thông tin như hiện nay, sản phẩm công nghệ ngày càng chịu sự đánh giá khắt khe hơn từ phía những người dùng, đặc biệt là về sản phẩm Game được nhận rất nhiều sự đánh giá từ phía các Game thủ, hay chỉ là những người chơi bình thường. Ngành công nghiệp Game hiện nay có thể nói là bùng nổ, với tốc độ phát triển đến chóng mặt, rất nhiều những Game hay và hấp dẫn đã được ra đời trong thời gian qua. Phía sau những Game phát triển và nổi tiếng như vậy đều có một Game Engine. Game Engine là một công cụ hỗ trợ, một Middleware giúp người phát triển viết Game một cách nhanh chóng và đơn giản, đồng thời cung cấp khả năng tái sử dụng các tài nguyên và mã nguồn cao do có thể phát triển nhiều Game từ một Game Engine.

Từ xu hướng phát triển và những bất cập trên, đồ án này sẽ khảo sát và nghiên cứu về Engine Unity – một Game Engine rất phổ biến và không kém mạnh mẽ hiện nay nhằm thực nghiệm việc phát triển một trò chơi (Demo) Adventure Rabbit 2D. Chuẩn bị kiến thức và kỹ năng cho định hướng nghề nghiệp (phát triển Game) sau này của chúng em, góp phần vào sự phát triển của ngành công nghiệp Game nước nhà. Các chương đầu trong tài liệu sẽ trình bày lần lượt các khái niệm chung về Game Engine và thế giới 2D.

Các chương tiếp theo sẽ giới thiệu về Engine Unity bao gồm các đặc điểm, tính năng, công cụ và thành phần trong Engine này. Sau những nội dung về thiết kế, tài liệu sẽ trình bày về việc lập trình trên Unity khi giới thiệu về ngôn ngữ lập trình cùng các lớp, hàm trong thư viện dựng sẵn của Unity thông qua các ví dụ thực tế khi phát triển một game Adventure Rabbit 2D. Cuối cùng là giới thiệu về bối cảnh, cốt truyện, tài liệu hướng dẫn sử dụng của trò chơi Demo cùng chương tổng kết về các khó khăn và những thành quả trong suốt quá trình phát triển trò chơi Demo trên Unity.

Dự án này sẽ là một cuộc phiêu lưu tuyệt vời. Em sẽ gặp phải những thách thức, học hỏi từ những sai lầm, và chia sẻ niềm vui trong mỗi bước tiến trình. Em mong rằng Thầy\Cô và bạn bè sẽ cùng với em theo dõi dự án này và chia sẻ những ý kiến đóng góp quý báu của mình để em có thể làm cho trò chơi này trở thành một tác phẩm hoàn hảo.

Cuối cùng, em muốn gửi lời cảm ơn đến tất cả những người ủng hộ em trong dự án này. Cùng nhau, chúng ta sẽ tạo ra một trò chơi đáng nhớ và thú vị. Hãy cùng em bắt đầu hành trình này và khám phá thế giới của trò chơi điện tử thông qua Unity.

# 1. Yêu cầu (Requirements)

## **1.1 Đặt vấn đề (Problem statement)**

### 1.1.1 Mô tả vấn đề

Xã hội ngày nay đã trở nên rất phụ thuộc vào sự hiện diện và vai trò của ngành công nghiệp game. Nếu một ngày không có game, cuộc sống của chúng ta sẽ trở thành một trải nghiệm khác biệt và có thể đầy thách thức. Các trò chơi điện tử không chỉ là một phương tiện giải trí, mà còn là một phần quan trọng của cuộc sống hàng ngày, ảnh hưởng đến cả khía cạnh văn hóa và xã hội.

Thứ nhất, việc thiếu game có thể dẫn đến sự trống rỗng trong thời gian rảnh rỗi của mọi người. Game đã trở thành một cách thường xuyên để thư giãn và giải tỏa căng thẳng, và nếu thiếu chúng, người dân sẽ phải tìm kiếm các hình thức giải trí khác, có thể không hiệu quả bằng game

Thứ hai, ngành công nghiệp game đang cung cấp một nguồn thu nhập lớn cho nhiều người, từ nhà phát triển game đến các nghệ sĩ, thiết kế viên đồ họa, và nhiều công việc liên quan khác. Sự thiếu hụt game có thể ảnh hưởng đến nền kinh tế và việc làm của nhiều người.

Thứ ba, game cũng đóng vai trò quan trọng trong việc kết nối xã hội. Nhiều trò chơi có tính năng đa người chơi trực tuyến, cho phép người chơi kết nối và tương tác với nhau từ xa. Thiếu game có thể tạo ra cảm giác cô đơn và cách ly trong xã hội.

Nhưng cũng cần lưu ý rằng sự cân nhắc trong việc sử dụng game là quan trọng. Sự phụ thuộc quá mức vào game cũng có thể gây hại cho sức khỏe tinh thần và thể chất. Việc duy trì một sự cân bằng là quan trọng để đảm bảo rằng game chỉ đóng vai trò là một phần trong cuộc sống xã hội, không chiếm lấy toàn bộ thời gian và tâm trí của chúng ta

### 1.1.2 Bối cảnh xu hướng công nghệ hiện nay

**Bối cảnh:**

Thế giới đang trải qua một sự phát triển công nghệ đáng kinh ngạc trong những năm qua. Công nghệ đã thay đổi cách thức mọi người làm việc, giải trí và giao tiếp với nhau. Các công nghệ mới như trí tuệ nhân tạo (AI), thực tế ảo (VR), internet vạn vật (IoT), máy học (Machine Learning), và blockchain đang có ảnh hưởng đến mọi lĩnh vực của đời sống.

**Xu hướng công nghệ:**

1. Trí tuệ nhân tạo (AI): Trí tuệ nhân tạo là một lĩnh vực nghiên cứu cung cấp cho máy tính khả năng học tập và tự động thực hiện nhiều nhiệm vụ. AI đang được sử dụng trong nhiều lĩnh vực như bán lẻ, y tế, ngân hàng, vận chuyển, và sản xuất.
2. Thực tế ảo (VR): VR cho phép người dùng trải nghiệm một môi trường ảo như thật thông qua kính thực tế ảo. Nó đang được sử dụng trong giáo dục, giải trí, y tế, và ngành công nghiệp trò chơi điện tử.
3. Internet vạn vật (IoT): IoT là một mạng lưới các thiết bị kết nối với nhau, cho phép chúng ta quản lý và kiểm soát các thiết bị từ xa. IoT đang được sử dụng trong nhà thông minh, công nghiệp, năng lượng, và giao thông.
4. Máy học (Machine Learning): Máy học là một phương pháp cho phép máy tính học hỏi từ dữ liệu mà không cần được lập trình một cách tường minh. Máy học đang được sử dụng trong các ứng dụng như xử lý ngôn ngữ tự nhiên, phân tích hình ảnh, và dự báo dữ liệu.
5. Blockchain: Blockchain là một công nghệ cho phép các giao dịch được thực hiện một cách an toàn và minh bạch. Nó đang được sử dụng trong các lĩnh vực như tài chính, chứng khoán, và quản lý chuỗi cung ứng.

Trên thế giới, các công nghệ này đang phát triển rất nhanh, tạo ra những thay đổi đáng kể đối với cuộc sống của mọi người. Việc đón đầu và ứng dụng những công nghệ mới sẽ giúp cải thiện hiệu quả và tăng tính cạnh tranh cho các doanh nghiệp trong nhiều lĩnh vực, trong đó có lĩnh vực trò chơi điện tử.

Lĩnh vực trò chơi điện tử đang phát triển rất nhanh với sự ra đời của các công nghệ mới như VR và AI, tạo ra những trải nghiệm chơi game mới và đa dạng hơn cho người chơi. Các công nghệ này giúp tạo ra những môi trường chơi game ảo thực tế và đưa người chơi vào trải nghiệm tuyệt vời như tham gia vào thế giới ảo hoàn toàn mới, mang đến những cảm xúc tuyệt vời cho người chơi.

Ngoài ra, sự phát triển của công nghệ Machine Learning cũng đang giúp cho các trò chơi điện tử có thể đáp ứng nhu cầu của người chơi một cách chính xác hơn. Việc phân tích và thu thập dữ liệu của người chơi giúp cho các nhà sản xuất trò chơi có thể cải thiện và điều chỉnh trò chơi theo nhu cầu của người chơi, từ đó nâng cao trải nghiệm chơi game và tạo ra một sản phẩm tốt hơn.

Từ đó, có thể thấy rằng việc ứng dụng các công nghệ mới trong lĩnh vực trò chơi điện tử sẽ giúp cho các doanh nghiệp trong lĩnh vực này tăng cường tính cạnh tranh, tạo ra sản phẩm tốt hơn và đáp ứng nhu cầu của người chơi một cách tốt nhất.

Tuy nhiên không phải ai cũng có quá nhiều thời gian hay không quá am hiều về công nghê sẽ rất khó có thể tiếp cận được với các xu hướng công nghệ hiện đại ngày nay, họ sẽ cần đến những thiết bị giải trí đơn giản để vừa có thể giảm căng thẳng mà không mất quá nhiều thời gian tìm hiểu về thiết bị giải trí hiện ngày nay.

### 1.1.3 Các giải pháp đã có

Game chủ yếu tập trung vào 1 nhân vật là chú thỏ tên Mon của dòng 2d platformer hay còn gọi là game đi màn vượt chướng ngại vật trên đường đi của nhân vật và qua nhiều ải với độ khó dễ khác nhau. Lối chơi của các game này liên quan chủ yếu tới việc di chuyển nhân vật phải có 1 số kĩ năng nhất định để có thể vượt qua các chướng ngại vật ví dụ như dụng vào gai nhân vật sẽ bị chết sẽ quay lại vị trí xuất phát của màn chơi hiên tại, người chơi phải thật sự cẩn thận và có sự tập trung cao độ trong các bước di chuyễn mỗi màn chơi đều có rất nhiều bẫy đặt sẵn và phải tư duy để phát hiện bẫy. Qua mỗi màn số lượng bẫy cản trở sẽ càng tăng lên tạo cho người chơi sự ức chế, cũng như hứng thú khi muốn vượt qua cho được màn chơi đó. Game được phát triển dựa tên ý tưởng của game Mario huyền thoại, nên sẽ tạo nên sự thân thuộc và dễ hòa nhập với cách chơi mà game đã quy định.

### 1.1.4 Các giải pháp đề xuất

- Adventure rabbit có thể kết hợp các yếu tố các giải pháp đã có và thêm các điểm nổi bật sau:

* Hình ảnh và âm thanh độc đáo: sử dụng đồ họa và âm thanh chất lượng cao để tạo ra một thế giới hấp dẫn cho người chơi, với các điểm riêng biệt mô phỏng 1 thế giới kì diệu của Rabbit.
* Câu chuyện hấp dẫn: xây dựng ra một câu chuyện đầy người mê hoặc những tình tiết bất ngờ và nhân vật phong phú để kích thích tính tò mò của người chơi.
* Độ khó linh hoạt: cho phép người chơi lựa chọn mức độ khó khác nhau của trò chơi, phù hợp với người chơi mới và người chơi có kinh nghiệm.
* Chế độ chơi đa dạng: cung cấp nhiều chế độ chơi khác nhau đáp ứng mức độ đa dạng cho người chơi.

### 1.1.5 Cốt truyện của Game

Có một chú thỏ tên là Mon sống trong một bầy đàn hạnh phúc ở rừng sâu. Mon là một con thỏ nhỏ bé nhưng rất tinh nghịch và hiếu động. Mỗi ngày, Mon cùng các bạn thỏ chơi đùa, hái cỏ và chia sẻ những câu chuyện vui vẻ về cuộc sống của họ. Cuộc sống trong bầy đàn này thật ấm áp và an lành.

Nhưng một ngày nọ, trong một trò chơi trốn tìm, Mon bất ngờ bị lạc khỏi bầy đàn của mình. Khi Mon tỉnh dậy sau giấc ngủ ngắn, chú thỏ đang ở một nơi hoàn toàn xa lạ. Mọi thứ xung quanh đều âm u và tối tăm, không có mùa xuân xanh, không có tiếng cười của bạn bè thân thiết. Mon đứng đó, đối mặt với một thế giới hoàn toàn mới mẻ và đầy rẫy sự bí ẩn.

Trong thời gian dấn thân vào cuộc hành trình tìm đường về nhà, Mon phải đối mặt với nhiều thử thách và nguy hiểm. Chú thỏ nhỏ bé phải vượt qua rừng sâu rậm rạp, trèo qua các vách đá nguy hiểm và tìm đường qua các con sông nước lớn. Trong những đêm tối lạnh giá, Mon học cách tự bảo vệ mình bằng cách làm tổ từ cỏ và lá để tránh mưa và gió. Nó phải tìm kiếm thức ăn trong tự nhiên, học cách ẩn mình khỏi các con thú săn mồi đáng sợ.

Trong suốt hành trình dài hơi, Mon đã cảm nhận sự cô đơn khi không có bạn bè cùng chia sẻ niềm vui và khó khăn. Nhưng chú thỏ thông minh này không bao giờ từ bỏ. Mon đã học cách kiên nhẫn và tự tin, và ngày qua ngày, nó trở nên mạnh mẽ hơn. Những kỷ niệm về bầy đàn và những trò chơi vui vẻ đã giúp Mon vượt qua những thời điểm khó khăn nhất.

Cuối cùng, sau hàng tháng đi lang thang trong rừng sâu, Mon đến gần nhà của mình. Nhưng trong lúc tưởng rằng mình đã vượt qua mọi khó khăn, chú thỏ bất ngờ bị rơi vào một hố sâu. Mon rơi xuống trong sự hoảng sợ và bất lực, với mọi thứ xung quanh trở nên mờ đi và tiếng ồn ào biến mất.

Khi Mon tỉnh dậy, chú thỏ nhớ lại mọi chặng đường mình đã đi, mọi thử thách đã vượt qua và mọi niềm vui đã trải qua. Nhưng mọi thứ dường như chỉ là một giấc mơ. Mon nhận ra rằng cuộc hành trình vẫn chưa kết thúc, và nhiệm vụ của nó là tiếp tục tìm đường về nhà.

Và với quyết tâm và kiên nhẫn, Mon tiếp tục cuộc hành trình của mình, không ngừng khám phá và đối mặt với mọi thử thách trên con đường trở về với bầy đàn yêu quý của mình. Cho dù có những lúc khó khăn và cô đơn, Mon luôn biết rằng mình không bao giờ đơn độc trong cuộc hành trình này, vì trái tim nó luôn đầy ắp niềm hy vọng và dũng khí. Mon học được rằng cuộc sống đầy rẫy những chặng đường bất ngờ, và việc quyết định tiếp tục bước đi là điều quan trọng nhất.

# 2. Thiết kế và triển khai

## **2.1 Các yêu cầu chức năng**

- Trong game 2D, có một số tác nhân chính mà người chơi sẽ gặp phải trong quá trình chơi. Dưới đây là một số tác nhân phổ biến trong trò chơi này:

1. Adventure rabbit (người chơi): Rabbit là nhân vật chính trong trò chơi, và người chơi sẽ điều khiển chú thỏ có khả năng di chuyển, nhảy và tránh các chướng ngại vật để hoàn thành các cấp độ và đạt được mục tiêu cuối cùng là đến vạch đích.

2.Chướng ngại vật: Trong quá trình di chuyển qua các cấp độ, Adventure rabbit sẽ phải đối mặt với các chướng ngại vật như đinh, chướng ngại vật di động và cạm bẫy. Người chơi cần có kỹ năng và phản ứng nhanh để tránh chúng và tiến tới đích.

3.Cấp độ và môi trường: Trò chơi Adventure rabbit 2D có nhiều cấp độ và môi trường khác nhau để khám phá, bao gồm rừng, sa mạc, bóng đêm,…. Mỗi cấp độ có thể có thiết kế, địa hình và thử thách riêng, đòi hỏi người chơi phải khéo léo và sáng tạo để vượt qua.

## **2.2 Các yêu cầu phi chức năng**

- Hiệu suất ổn định: Hệ thống/game cần đảm bảo hiệu suất ổn định để tránh các lỗi hoặc giật lag khi người chơi trải nghiệm game.

- Hỗ trợ đa nền tảng: Game cần có khả năng hoạt động trên nhiều nền tảng khác nhau như điện thoại di động, máy tính bảng, và máy tính cá nhân.

- Ngôn ngữ và đa vùng: Hỗ trợ nhiều ngôn ngữ và địa điểm để mở rộng thị trường tiềm năng và thu hút đối tượng người chơi rộng rãi.

- Bảo mật và quyền riêng tư: Bảo vệ thông tin cá nhân của người chơi và đảm bảo tính bảo mật trong trò chơi.

- Giao diện người dùng thân thiện: Tạo ra một giao diện người dùng dễ sử dụng và hấp dẫn để tăng cường trải nghiệm người chơi.

- Hỗ trợ âm thanh và hình ảnh: Đảm bảo rằng game có âm thanh và hình ảnh chất lượng để tạo ra trải nghiệm thú vị.

- Tài liệu hướng dẫn và hỗ trợ: Cung cấp tài liệu hướng dẫn và hỗ trợ trực tuyến cho người chơi để họ có thể tận hưởng trò chơi một cách dễ dàng và thuận tiện.

- Cập nhật và bảo trì: Đảm bảo tính liên tục của game bằng cách cung cấp các bản cập nhật và dịch vụ bảo trì thường xuyên để sửa lỗi và thêm tính năng mới.

## **2.3 Các ràng buộc**

### 2.3.1 Các ràng buộc về triển khai

- Nền tảng và thiết bị: Trò chơi hoạt động trên các nền tảng và thiết bị cụ thể (ví dụ: iOS, Android, máy tính cá nhân).

- Kích thước tải về: Trong trường hợp của các ứng dụng di động, cần xem xét kích thước tải về để đảm bảo rằng trò chơi có thể được tải và cài đặt dễ dàng trên các thiết bị có dung lượng bộ nhớ hạn chế.

- Hạn chế về mạng: Trò chơi có thể phải hoạt động một cách ổn định và có trải nghiệm tốt dù người chơi có hoặc không có kết nối internet. Điều này có thể đặt ra ràng buộc về việc lưu trữ dữ liệu và chế độ chơi ngoại tuyến.

### 2.3.2 Các ràng buộc về kinh tế

- Ngân sách phát triển: để phát triển trò chơi đi lên thì cần đảm bảo tất cả các yêu cầu chức năng nằm trong danh sách giới hạn ngân sách.

- Mô hình kinh doanh: cần xác định mô hình kinh doanh về trò chơi.

- Phí duy trì và bảo trì: có kế hoạch rõ ràng cho việc bảo trì và cập nhập khi trò chơi Adventure rabbit sau khi trò chơi được phát hành

**2.3.3 Các ràng buộc về đạo đức**

- Quyền riêng tư: bảo vệ thông tin cá nhân của người chơi và đảm bảo về dữ liệu cá nhân người dùng.

- Nội dung phù hợp: đảm bảo nội dung  phù hợp với lứa tuổi và tuân theo các quy tắc hướng dẫn đạo đức.

## **2.4. Mô hình hệ thống / Thiết kế giải pháp**

### 2.4.1. Các kịch bản của hệ thống

1. **Kịch bản Bắt đầu Trò chơi:**
   1. Người chơi bắt đầu trò chơi bằng cách nhấn vào nút "Menu" trên màn hình chính.
   2. Hệ thống khởi đầu trò chơi bằng cách tạo ra một phiên bản của lớp **Game**.
   3. Người chơi ấn tiếp nút “Play” , vào giao diện chơi game sẽ điều khiển nhân vật thỏ Mon bằng cách sử dụng các nút hoặc cử chỉ trên màn hình.
2. **Kịch bản Di chuyển và Nhảy:**
   1. Trong quá trình chơi, người chơi có thể di chuyển Alex lên, xuống và nhảy để tránh chướng ngại vật và thu thập đồng xu.
   2. Hệ thống phải theo dõi các hành động này và cập nhật vị trí của thỏ Mon trên màn hình.
3. **Kịch bản Kiểm tra Va chạm:**
   1. Hệ thống phải thường xuyên kiểm tra xem Mon có va vào chướng ngại vật nào không.
   2. Nếu có va chạm, hệ thống phải kết thúc trò chơi và hiển thị điểm số cao nhất của người chơi trong lần chơi gần nhất.
4. **Kịch bản Kết thúc Trò chơi:**
   1. Trò chơi có một kịch bản kết thúc khi người chơi thua cuộc hoặc chọn kết thúc trò chơi.
   2. Hệ thống phải lưu điểm số cao nhất của người chơi để hiển thị trên bảng xếp hạng.
5. **Kịch bản Thoát trò chơi:**

Người chơi có thể thoát ra khỏi trò chơi bất cứ lúc nào bằng cách chọn tùy chọn "Thoát" trên màn hình chính.

### 2.4.2 Mô hình Use-case (Use-case)

A diagram of a game

Description automatically generated

### 2.4.3 Mô hình lớp và đối tượng

**1. Lớp Game**:

a. Đối tượng **Game** đại diện cho trò chơi tổng thể.

b. Có phương thức **start()** để khởi đầu trò chơi.

c. Có phương thức **end()** để kết thúc trò chơi khi người chơi thua cuộc.

**2. Lớp Player (Người chơi):**

a. Đối tượng **Player** đại diện cho nhân vật Alex.

b. Có thuộc tính score để lưu điểm số của người chơi.

c. Có phương thức **moveUp(), moveDown(),** và **jump()** để điều khiển Alex di chuyển lên, xuống và nhảy.

**3. Lớp Obstacle (Chướng ngại vật):**

a. Đối tượng **Obstacle** đại diện cho các chướng ngại vật trong trò chơi như cây cột, đá, hoặc con thú hoang dại.

b. Có thuộc tính **position** để xác định vị trí của chướng ngại vật trên màn hình.

c. Có phương thức **checkCollision(Player)** để kiểm tra xem Alex có va vào chướng ngại vật không.

**4. Lớp MainMenu (Màn hình chính):**

a. Đối tượng **MainMenu** đại diện cho màn hình chính của trò chơi.

b. Có các phương thức để hiển thị tùy chọn như "Bắt đầu", "Bảng xếp hạng", và "Thoát".

c. Có phương thức để chuyển đổi giữa các màn hình trò chơi khác nhau.

### 2.4.4 Các biểu đồ tuần tự

A diagram of a game

Description automatically generated

### 2.4.5 Các màn hình giao diện người dùng:

**Main chính:**

****

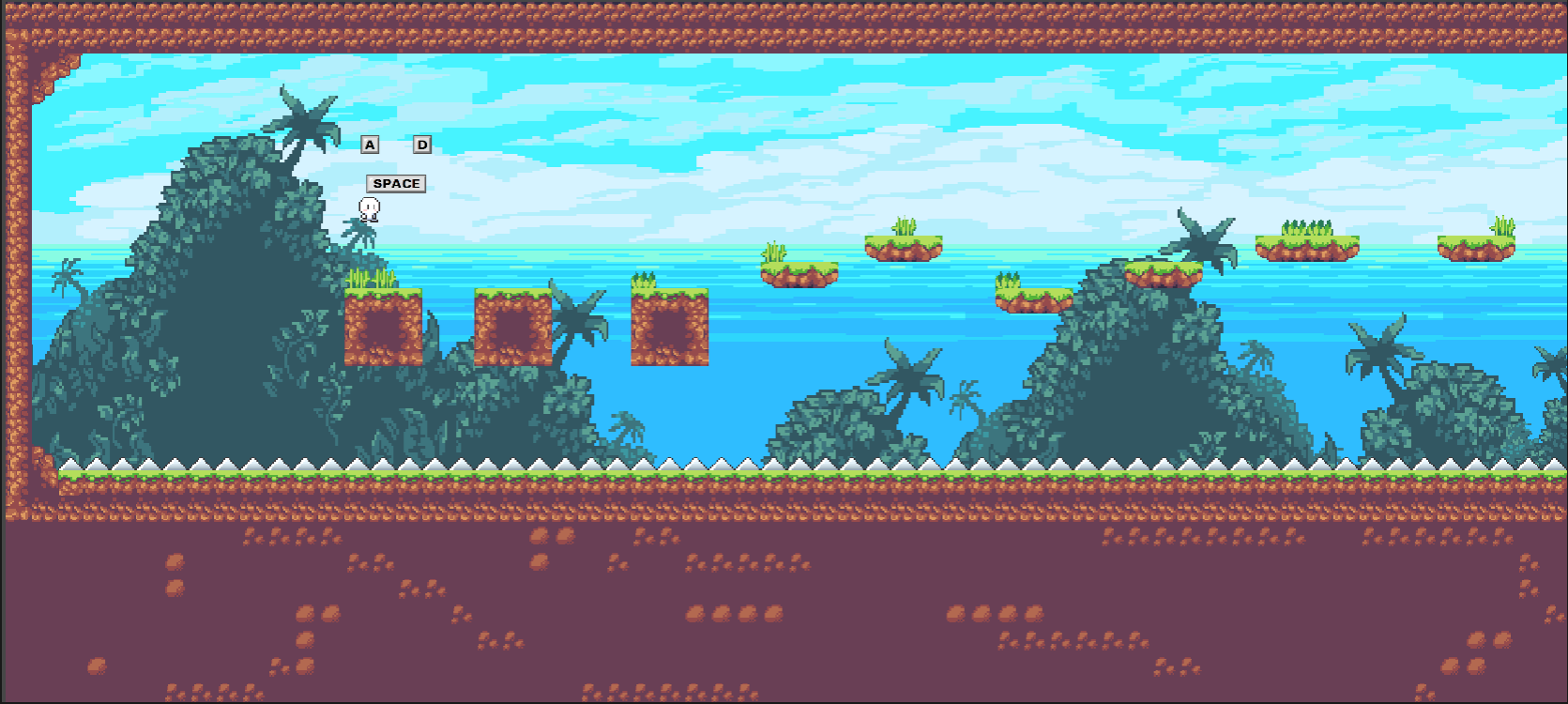
**Bảng Option**

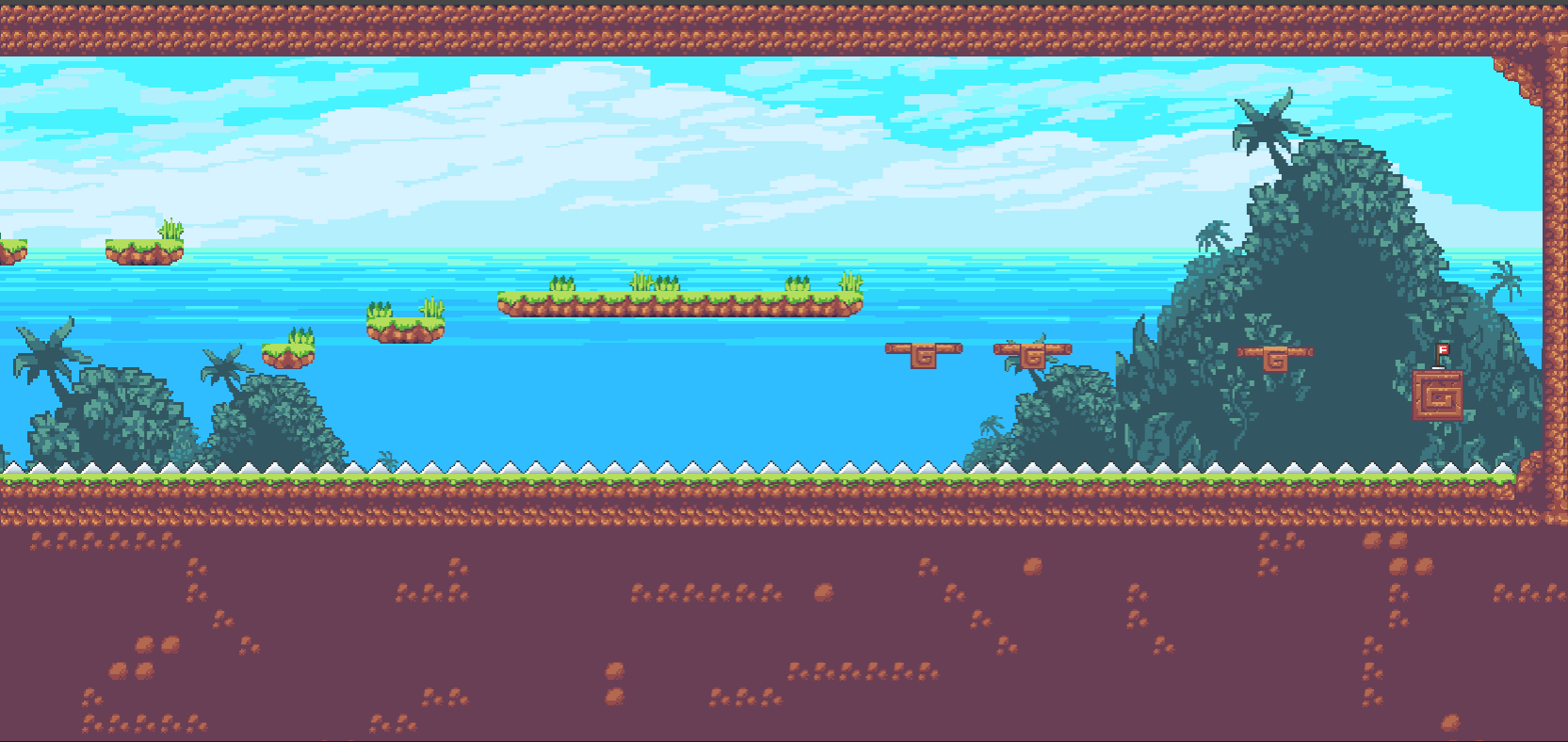
****

**Bảng Pause**

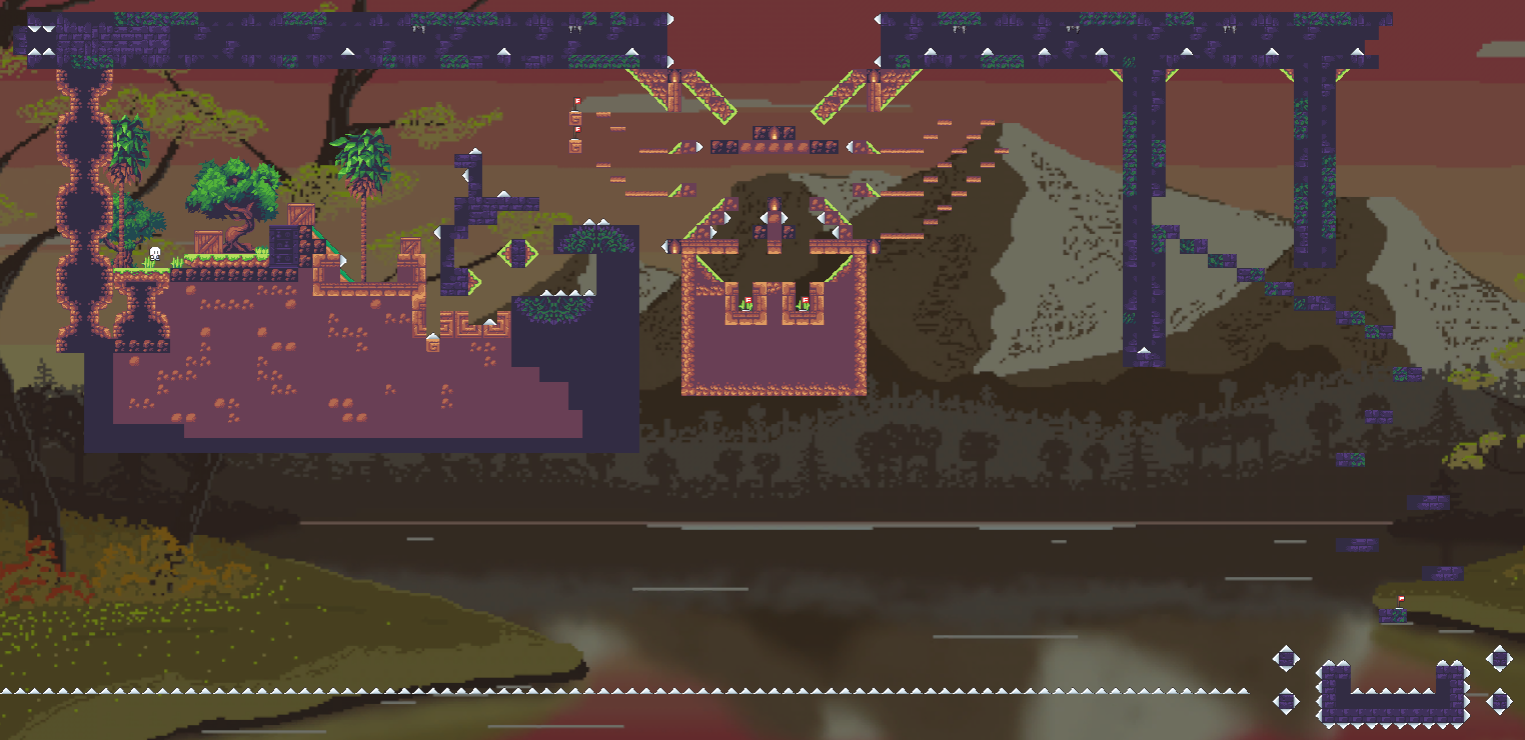
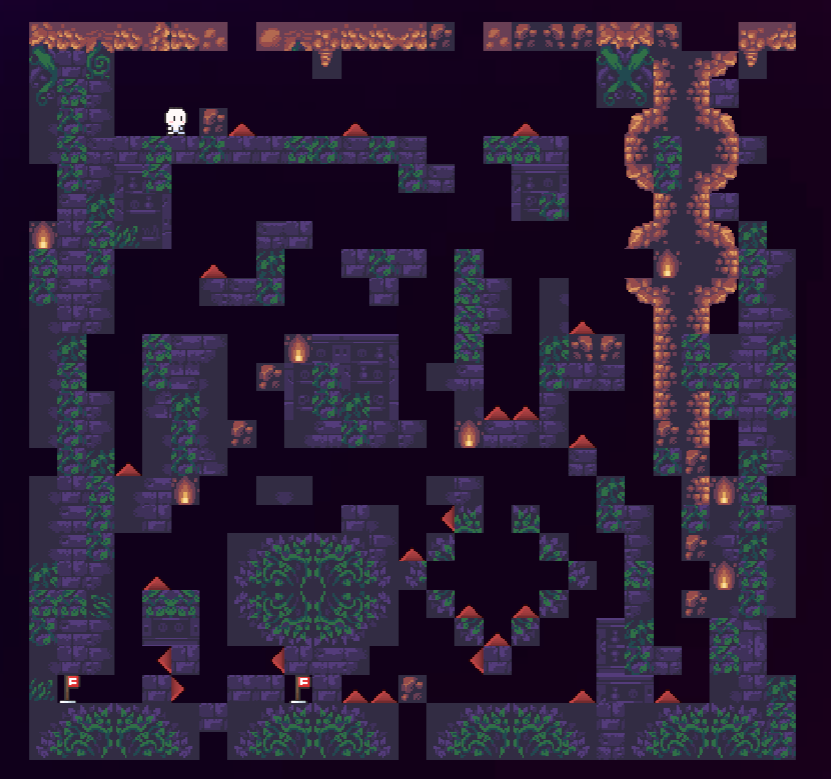
****

**Level 1**

****

****

**Level 2**

**Level 3Level 4**

**Lever 5**

****

**Level 6**

**Bảng Game Over**

### 2.4.6 Nhân vật và thiết kế lối chơi

**Player**

\*Player



**Bản đồ**

Tạo dựng bằng cách kết hợp nhiều cây cỏ, tường, khối ô vuông

A video game with a pixelated image

Description automatically generated

Di chuyển và chạy nhảy để tránh chướng ngại vật

A video game with a purple background

Description automatically generated

**Xử lý di chuyển, va chạm, kết thúc màn chơi**

Di chuyển: người chơi sử dụng bàn phím để di chuyển trái phải ngoài ra nhấn nút Space để nhảy.

Vật cản: : tạo vật cản tĩnh có chức năng nếu chạm vào sẽ bị chết.

A screenshot of a video game

Description automatically generated

**Tiến trình game**

 Người chơi phải khéo léo né các vật cản trên nền cỏ và đường đi.

Từ giao diện chính người chơi có những lựa chọn sau:

* Play – Bắt đầu trò chơi
* Options – Điều chỉnh thông số của các chức năng
* Quit – Thoát game.

# 3. Ngôn ngữ và công nghệ sử dụng

## **3.1 Giới thiệu về ngôn ngữ C#**

C# (đọc là "C thăng" hay "C sharp" )là một ngôn ngữ lập trình hướng đối tượng được phát triển bởi Microsoft, là phần khởi đầu cho kế hoạch .NET của họ. Tên của ngôn ngữ bao gồm ký tự thăng theo Microsoft nhưng theo ECMA là C#, chỉ bao gồm dấu số thường. Microsoft phát triển C# dựa trên C++ và Java. C# được miêu tả là ngôn ngữ có được sự cân bằng giữa C++, Visual Basic, Delphi và Java.

C# được thiết kế chủ yếu bởi Anders Hejlsberg kiến trúc sư phần mềm nổi tiếng với các sản phẩm Turbo Pascal, Delphi, J++, WFC.

C# theo một hướng nào đó, là ngôn ngữ lập trình phản ánh trực tiếp nhất đến.NETFramework mà tất cả các chương trình.NET chạy, và nó phụ thuộc mạnh mẽ vào Framework này. Mọi dữ liệu cơ sở đều là đối tượng, được cấp phát và hủy bỏ bởi trình dọn rác Garbage-Collector (GC), và nhiều kiểu trừu tượng khác chẳng hạn như class, delegate, interface, exception, v.v, phản ánh rõ ràng những đặc trưng của.NET runtime.

So sánh với C và C++, ngôn ngữ này bị giới hạn và được nâng cao ở một vài đặc điểm nào đó, nhưng không bao gồm các giới hạn sau đây:

- Các con trỏ chỉ có thể được sử dụng trong chế độ không an toàn. Hầu hết các đối tượng được tham chiếu an toàn, và các phép tính đều được kiểm tra tràn bộ đệm. Các con trỏ chỉ được sử dụng để gọi các loại kiểu giá trị; còn những đốitượng thuộc bộ thu rác (garbage-collector) thì chỉ được gọi bằng cách tham chiếu.

- Các đối tượng không thể được giải phóng tường minh

- Chỉ có đơn kế thừa, nhưng có thể cài đặt nhiều interface trừu tượng (abstractinterfaces). Chức năng này làm đơn giản hóa sự thực thi của thời gian thực thi.

- C# thì an-toàn-kiểu (typesafe) hơn C++.

- Kiểu thứ tự được thay thế bằng tên miền không gian (namespace).

- C# không có tiêu bản.

- Có thêm Properties, các phương pháp có thể gọi các Properties để truy cập dữ liệu.

- Có reflection.

## **3.2 Công cụ sử dụng**

### 3.2.1 Unity 2023.1.0f1

Unity2D là phần mềm làm games trực tiếp theo thời gian thực, mà không cần render, cho phép người design game có thể thiết kế InterfaceGraphic, map hay character … từ một phần mềm thứ 2 (thường là các phần mềm thiết kế đồ họa chuyên nghiệp như 3Dsmax, Blender, Maya, XSL,Cinema4D,Cheetah3D, Modo, Autodesk FBX, LightWave…) sau đó chỉ việc import nó vào trong Unity với định dạng của tập tin là \*.FBX hay \*.dae, \*.3DS, \*.dxf và \*.obj, nhưng định dạng \*.FBX hay được dùng vì được tối ưu hóa hơn, còn dùng các định dạng khác ngoài FBX thì phải cài phần mền thiết kế character tưng ứng thì mới dùng được (tức là dùng Maya có định dạng \*.mb, \*.ma thì muốn dùng được phải cài Maya).

Ngoài ra khi bạn design được một game thì bạn có thể xuất được ra một file.exe vàcó thể chạy và chơi được trên PC khác.

Một thế mạnh nữa của Unity là bạn có thể chạy demo game của bạn ngay trong khi design, nó có hỗ trợ hai chế độ là Scene và Game, rất thuận tiện cho việc test thử các modul Game.

Unity2D có 2 loại phiên bản, một cho người dùng free, và một phiên bản pro thì mất phí.

Khi tải Unity về các bạn cài đặt bình thường, đến khi kết thúc cài đặt nền các bạn chọn phiên bản Pro thì cần phải mua, còn nếu là người dùng free thì các bạn chọn kích hoạt qua web, chỉ cần có kết nối mạng internet, sau đó chọn cài đặt thủ công và nhập mail, chọn free là bạn có thể sử dụng một bản Unity free.

Unity hỗ trợ 3 ngôn ngữ phát triển chính là C#, UnityScript (dựa trên Javascript) và Boo. Boo là ngôn ngữ ít được sử dụng nhất và thường được khuyên không nên dùng. C# là ngôn ngữ được dùng phổ biến nhất trong cộng đồng Unity với rất nhiều plugin và ví dụ sử dụng ngôn ngữ này. Bạn có thể cân nhắc sử dụng Unity script tuy nhiên có thể bạn sẽ gặp vấn đề khi tương tác với một số plugin được viết bằng C#.

### 3.2.2 Ưu điểm và nhược điểm

**ϖƯu điểm:**

- Unity có rất nhiều tính năng tuyệt vời:

- Unity có một cồng đồng rất lớn về asset và plugin – trong đó có rất nhiều

resources free và có nhiều thứ rất đáng bỏ tiền.

- Unity có bộ công cụ rất trực quan và editor có thể mở rộng bằng plugins.

- Unity hỗ trợ rất nhiều định dạng asset khác nhau và có thể tự động chuyển đổi đến định dạng phù hợp nhất với nền tảng thích hợp.

- Unity hỗ trợ nhiều nền tảng: di động, desktop, web và console.

- Việc triển khai đến các nền tảng khác nhau cũng khá dễ quản lý.

- Bạn có thể dễ dàng xây dựng một game 2D mà không cần cấu hình quá phức tạp.

- Unity bản free có hầu hết những tính năng quan trọng nhất.

- Unity bản trả phí phù hợp với các developer chuyên nghiệp

**ϖNhược điểm:**

- Tuy nhiên Unity cũng có vài nhược điểm mà bạn cần cân nhắc:

- Việc hợp tác rất khó khăn. Unity sử dụng một server asset rất hiệu quả để hỗ trợ các đội phát triển phần mềm hợp tác với nhau. Tuy nhiên nếu bạn không sử dụng nó thì việc chia sẽ code và asset giữa các thành viên trong team có thể gây ra những vấn đề nghiêm trọng. Lựa chọn tốt nhất là sử dụng một số công cụ quản lý resource bên ngoài nhưng có một vài binary file không thể merge được 19 với nhau và việc cập nhật asset có thể gây nên một số vấn đề trong scenes, mất kết nối đến script và các đối tượng khác.

- Hiệu năng chưa thật sự ấn tượng cho đến khi Unity 5 ra mắt. Unity 5 đã chạy hầu hết trên một luồng duy nhất và hầu như không sử dụng thêm 1 nhân phụ nào trên các thiết bị di động. Bộ biên dịch chưa được tối ưu tốt cho các bộ xử lý ARM trên hầu hết các thiết bị di động. Để giải quyết vấn đề này thì Unity đã quyết định transpile (source-to-source compiler) sang C++ và sử dụng LLVM để tối ưu được nhiều hơn thay vì giải quyết vấn đề này trực tiếp trên các phiên bản sau này.

- Mã nguồn của engine không được công bố kể cả cho những người dùng chấp nhận trả tiền. Điều đó có nghĩa là nếu bạn gặp một bug với engine bạn phải chờ Unity fix chúng trong các bản tiếp theo. Điều này có thể gây nên những vấn đề nghiêm trọng với project của bạn.

# 4 Tìm hiểu về unity

## **4.1 Unity là gì?**

Unity là một nền tảng phát triển trò chơi và ứng dụng đa nền tảng (cross-platform) phổ biến. Nó là một môi trường phát triển tích hợp (integrated development environment - IDE) được thiết kế để giúp các nhà phát triển tạo ra trò chơi, ứng dụng thực tế ảo (VR), thực tế ảo mở rộng (AR), và nhiều loại ứng dụng tương tự trên nhiều nền tảng khác nhau, bao gồm PC, điện thoại di động, máy tính bảng, máy chơi game và nhiều hệ điều hành khác.

Unity được phát triển bởi công ty Unity Technologies và cung cấp một loạt công cụ mạnh mẽ để tạo, chỉnh sửa và quản lý đồ họa, âm thanh, vật lý, và logic trò chơi. Nó hỗ trợ nhiều ngôn ngữ lập trình, bao gồm C#, JavaScript, và Boo, cho phép các nhà phát triển tạo ra trò chơi và ứng dụng đa dạng.

Unity cũng có cộng đồng lớn và hệ thống tài liệu phong phú, giúp người dùng học và chia sẻ kiến thức về phát triển trò chơi và ứng dụng. Unity đã được sử dụng rộng rãi trong ngành công nghiệp trò chơi và ứng dụng tương tác, từ phát triển trò chơi độc lập đến việc tạo các ứng dụng mô phỏng thực tế ảo và nhiều ứng dụng thú vị khác.

**4.2 Quá trình phát triển game Unity**

Unity là một nền tảng phát triển trò chơi và ứng dụng đã có một quá trình phát triển dài hơn một thập kỷ. Dưới đây là một số sự kiện quan trọng trong quá trình phát triển của Unity:

**1.** Ngày thành lập (2004): Unity Technologies được thành lập tại Copenhagen, Đan Mạch bởi David Helgason, Nicholas Francis, và Joachim Ante. Ban đầu, họ tập trung vào phát triển công nghệ đồ họa 3D cho trò chơi trực tuyến.

**2.** Ra mắt Unity 1.0 (2005): Unity phát hành phiên bản đầu tiên của nền tảng vào tháng 6 năm 2005, chủ yếu dành cho macOS. Phiên bản đầu tiên này đã thiết lập cơ sở cho việc phát triển trò chơi bằng Unity.

**3.** Phiên bản Unity 2.0 và 3.0 (2007 - 2009): Unity 2.0 và 3.0 đưa vào nhiều tính năng mạnh mẽ và cải tiến đáng kể. Unity bắt đầu hỗ trợ nhiều nền tảng và trở nên phổ biến hơn trong cộng đồng phát triển trò chơi.

**4.** Ra mắt Unity Asset Store (2010): Unity Asset Store ra mắt, cho phép các nhà phát triển mua và bán các tài nguyên như mô hình 3D, âm thanh, và cả mã nguồn, giúp tạo điều kiện thuận lợi cho việc phát triển trò chơi.

**5.** Unity 4 và 5 (2012 - 2015): Unity 4 và 5 đưa vào nhiều tính năng mạnh mẽ như hệ thống vật lý tích hợp, hỗ trợ nhiều nền tảng rộng rãi, và đồ họa cải tiến.

**6.** Ra mắt Unity Technologies Vietnam (2014): Unity mở một cơ sở phát triển tại Hà Nội, Việt Nam, để thúc đẩy sự phát triển và nâng cấp sản phẩm.

**7.** Phiên bản Unity 2017 (2017): Unity 2017 ra mắt với cải tiến đáng kể trong hiệu suất và hỗ trợ nhiều công nghệ mới như AR (thực tế ảo) và VR (thực tế ảo).

**8.** Unity 2018 và sau này (2018 - nay): Unity tiếp tục phát triển và cập nhật các phiên bản mới với nhiều cải tiến trong công nghệ đồ họa, hiệu suất, và hỗ trợ cho nền tảng mới. Unity đã trở thành một trong những nền tảng phát triển trò chơi và ứng dụng hàng đầu trên thị trường.

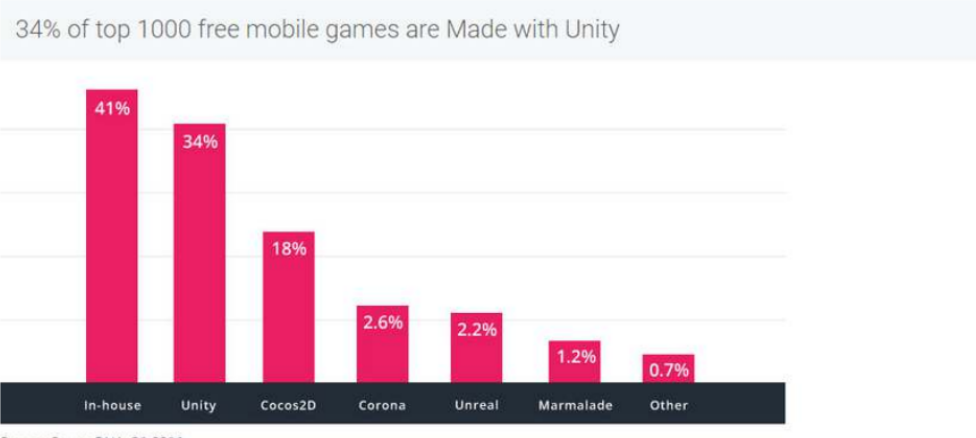
Trong quá trình phát triển này, Unity đã thu hút một cộng đồng lớn của các nhà phát triển trò chơi và ứng dụng trên toàn thế giới và đã đóng một vai trò quan trọng trong việc thúc đẩy sự phát triển của ngành công nghiệp trò chơi và thực tế ảo/ảo hóa.

## **4.3 Một số thống kê về Unity**

- Tính đến quý 3 năm 2016 đã có 5 tỉ lượt download game và ứng dụng được phát triển bởi Unity

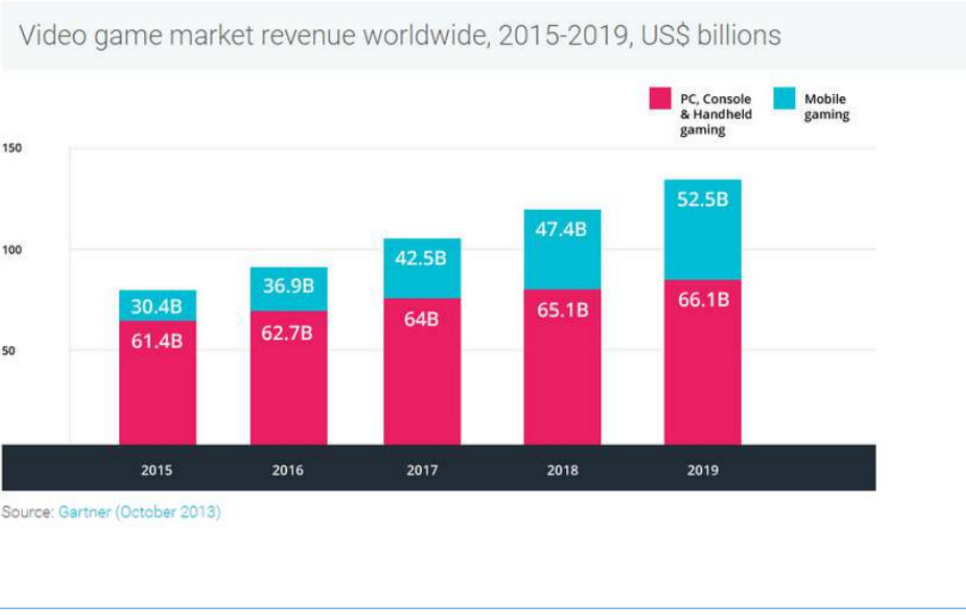
- 2,4 tỉ thiết bị di đô vng đã từng tải ít nhất 1 ứng dụng bởi unity.

- Trong top 1000 game Mobiles miễn phí thì số lượng game tạo ra bởi Unity chiếm tới 34%.



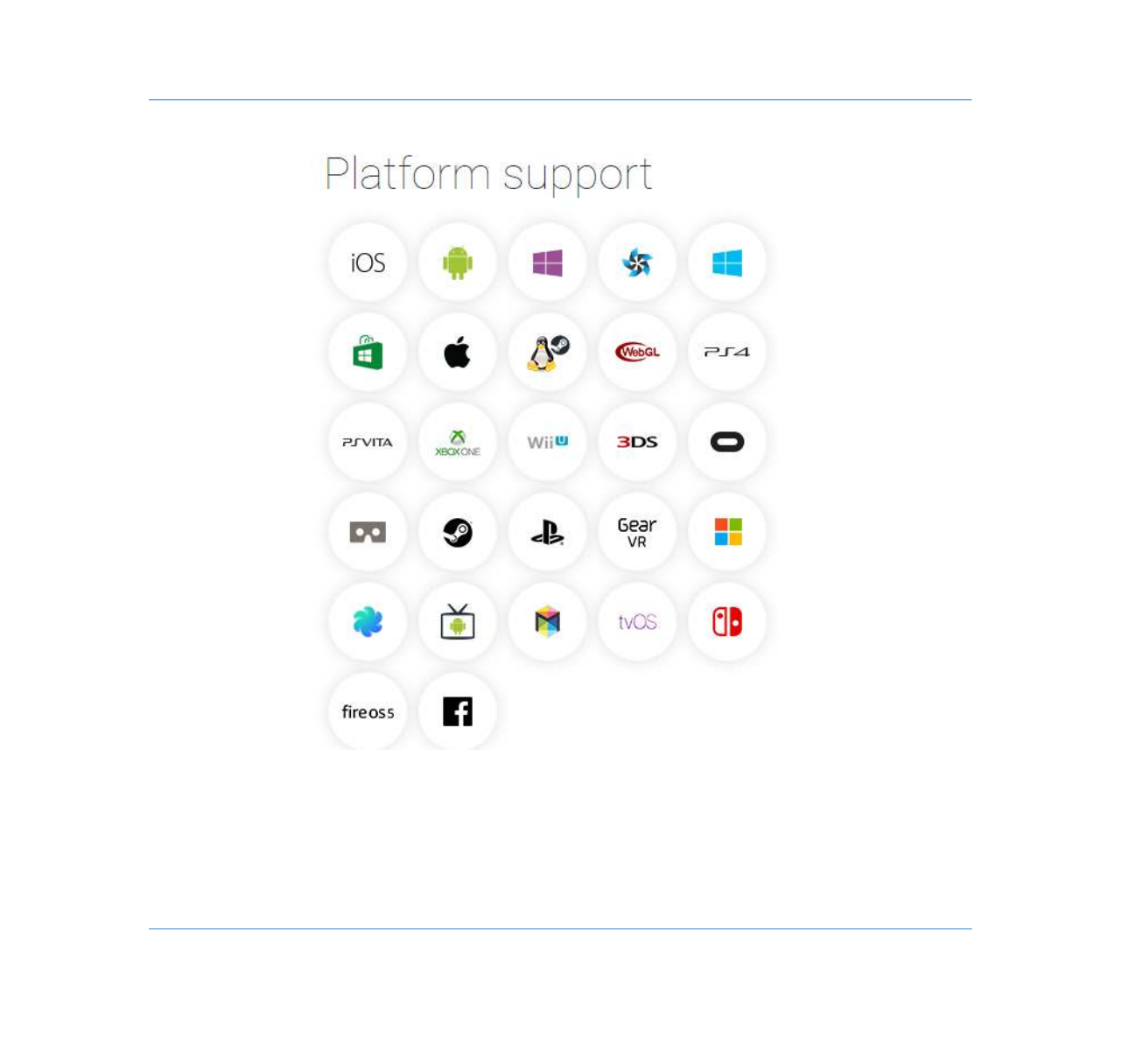
- Số lượng người dùng (gamer) của Unity đạt tới con số 770 triệu, trong khi đó số người thường xuyên sử dụng Twitter là 310 triệu người.

- Sự thay đổi trong cách thức chơi game của người chơi hay nói cách khác là xu hướng mọi người tập trung vào game trên di động nhiều hơn.



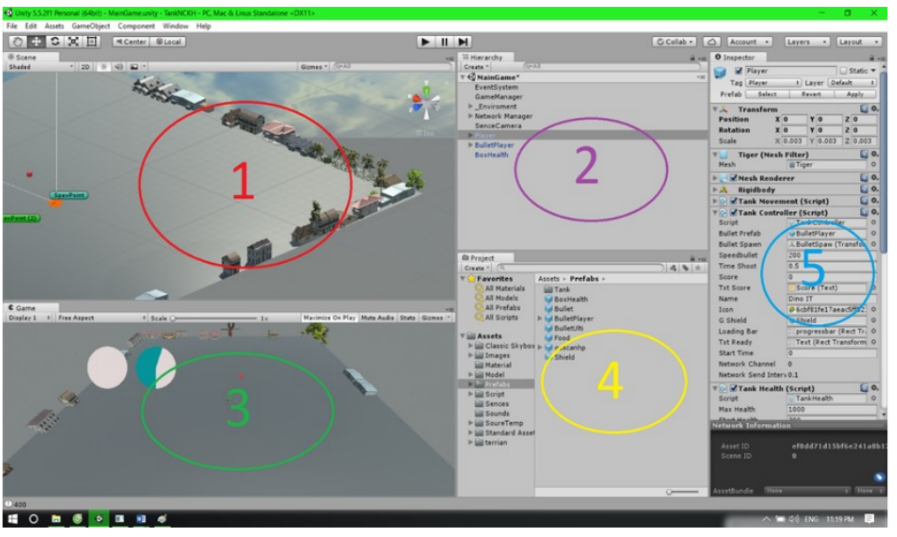
## **4.4 Ưu điểm của Unity**

- Chức năng cốt lõi đa dạng bao gồm: cung cấp công cụ dựng hình (kết xuất đồ họa) cho các hình ảnh 2D hoặc 3D, công cụ vật lý (tính toán và phát hiện va chạm), âm thanh, mã nguồn, hình ảnh động, trí tuệ nhân tạo, phân luồng, tạo dò ng dữ liệu xử lý, quản lý bộ nhớ, dựng ảnh đồ thị và kết nối mạng. Nhờ có các engine mà công việc làm game trở nên ít tốn kém và đơn giản hơn. Hỗ trợ đa nền tảng: Một trong các thế mạnh của Unity3D chính là khả năng 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. Nói cách khác, chỉ với một gói engine, các studio có thể làm game cho bất kỳ hệ điều hành nào và dễ dàng convert chúng sang những hệ điều hành khác nhau. Đồng thời, đây cũng là giải pháp cho các game online đa nền tảng – có thể chơi đồng thời trên nhiều hệ điều hành,phần cứng khác nhau như Web, PC, Mobile, Tablet….

- Dễ sử dụng: Unity3D được built trong một môi trường phát triển tích hợp, cung cấp một hệ thống toàn diện cho các lập trình viên, từ soạn thảo mã nguồn, xây dựng công cụtự động hóa đến trình sửa lỗi. Do được hướng đến đồng thời cả lập trình viên không chuyên và studio chuyên nghiệp, nên Unity3D khá dễ sử dụng. Hơn nữa, đây là một trong những engine phổ biến nhất trên thế giới, người dùng có thể dễ dàng tìm kiếm kinh nghiệm sử dụng của “tiền bối” trên các forum công nghệ.Tính kinh tế cao: Unity Technologies hiện cung cấp bản miễn phí engine Unity3D cho người dùng cá nhân và các doanh nghiệp có doanh thu dưới 100.000 USD/năm. Với bản Pro, người dùng phải trả 1.500 USD/năm – một con số rất khiêm tốn so với những gì engine này mang lại 

# 5. Tìm hiểu về unity engine

## **5.1 Các thành phần trong Unity Editor**

****

1. **Cửa sổ Sences**

- Phần này phần 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 ...

- Phần này có để thiết lập một số thông số như hiển thị ánh sáng, âm anh, 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 năngquan trọng nhất của Unity.

1. **Cửa sổ Hierarchy**

Tab hierarchy là nơi hiển thị các Game Object trong Sences hiện hành. Khi các đối tượng được thêm hoặc xóa trong Sences, 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.

1. **Cửa sổ Game**

Đây là mạn 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 man hình,thông số (stats), gizmos, tùy chọn bật tắt các component...

1. **Cửa sổ Project**

Đây là cưa 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.

1. **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 tuỳ 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, …

## **5.2 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...

1. **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.

1. **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.

1. **Animation**

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

1. **Key FrameKey**

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

1. **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 môt đối tượng ban đầu. Ví dụ: Các đối tượng là đồng tiên 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.

1. **Sounds**

Âm thanh trong game.

1. **Script**

Script là thành phân quan trọng nhất trong Unity, có thế xem scripts như là linh hồn của game. Chúng ta có thể viết kịch bản cho game bằng C#, Java Scripts hoặc Boo (một dẫn xuất của ngôn ngữ Python). Theo nhiều người đã sử dụng Unity thì code bằng C# sẽ giúp game chạy nhanh hơn và giúp kiểm soát code tốt hơn do tất cả các biến phải được khai báo rõ ràng. Mặt khác ngôn ngữ C# rất tiện dụng để lập trình, nên trong thành phẩm Game 2D chúng em dùng ngôn ngữ C# để viết kịch bản cho game. Mỗi file script C# là một class bắt buộc kế thừa từ lớp MonoBehaviour, có tên class phải trùng với tên file script.

Một đoạn script muốn thực thi được thì nó phải được gắn vào một đối tượng.

1. **Scenes**

Trong Unity, chúng ta có thể xem Scenes là các màn chơi, cấp độ chơi riêng lẻ, hoặc các vùng của nội dung trò chơi. Ví dụ như Main menu. Options, About

Bằng cách xây dựng trò chơi với nhiều cảnh, chúng ta sẽ có the phân phối thời gian tài và thử nghiệm các phân khác nhau của trò chơi riêng lẻ mot cách nhanh chóng và chính xác.

1. **Assets**

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

1. **Camera**

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

1. **Transform**

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

# 6. Một số thành phần khác của dự án

## **6.1 Kế hoạch dự án**

1. **Xác định Mục tiêu và Phạm vi Dự án**

- Xác định mục tiêu chính của trò chơi: Thể loại, người chơi mục tiêu, trải nghiệm mong muốn.

**-**  Xác định phạm vi dự án: Xác định loại trò chơi, số màn chơi, tính năng cơ bản, và nội dung cần thiết.

1. **Tạo Thiết kế Cơ bản**

-Tạo bản thiết kế cơ bản cho trò chơi: Vẽ bản đồ, xác định nhân vật và đối tượng, và xây dựng cấu trúc chung của trò chơi.

1. **Lập Kế Hoạch Phát Triển**

- Xác định cấu trúc dự án: Tạo cấu trúc thư mục, quản lý tài nguyên (âm thanh, hình ảnh, v.v.), và sắp xếp mã nguồn.

- Xác định mô hình phát triển: Xác định thời gian và trình tự phát triển, với các giai đoạn và mốc thời gian.

1. **Phát Triển**

- Tạo cơ sở hạ tầng: Tạo cơ sở hạ tầng cơ bản của trò chơi, bao gồm các hệ thống quản lý màn chơi, nhân vật, và đối tượng.

- Phát triển tính năng chính: Thêm các tính năng cơ bản như di chuyển nhân vật, xử lý va chạm, và hệ thống điều khiển.

- Tạo nội dung: Tạo màn chơi, đối tượng, hình ảnh, âm thanh và hiệu ứng đặc biệt.

1. **Kiểm Tra và Sửa Lỗi**

- Kiểm tra và sửa lỗi: Thực hiện kiểm tra chất lượng và kiểm tra tích hợp để tìm và sửa lỗi.

1. **Tối ưu hóa và Hiệu suất**

- Tối ưu hóa hiệu suất: Cải thiện hiệu suất trò chơi bằng cách tối ưu hóa mã nguồn, sử dụng cụ thể, và kiểm tra trên nhiều thiết bị.

1. **Kiểm Tra Cuối cùng và Phát Hành**

- Kiểm tra cuối cùng: Kiểm tra cuối cùng để đảm bảo rằng trò chơi hoạt động một cách ổn định và thỏa mãn mục tiêu.

- Phát hành: Chuẩn bị trò chơi cho việc phát hành trên các nền tảng mong muốn (PC, di động, v.v.).

1. **Hỗ Trợ và Cập Nhật**

-  Hỗ trợ và duy trì: Xây dựng hệ thống hỗ trợ người chơi và lập kế hoạch cho việc cập nhật trò chơi sau khi phát hành.

## **6.2 Các vấn đề đạo đức và làm việc chuyên nghiệp**

**- Vấn đề đạo đức:**

* Trung thực và Thành thật: Luôn nói sự thật và không che giấu thông tin quan trọng. Tránh gian lận hoặc làm việc không trung thực.
* Bảo vệ thông tin riêng tư: Bảo vệ thông tin cá nhân của người khác và không tiết lộ thông tin cá nhân mà không có sự đồng ý.

**- Vấn đề làm việc chuyên nghiệp:**

* Sự Tận Tâm: Làm việc với tận tâm và cam kết đối với công việc và dự án của bạn.
* Chuyên Môn: Phát triển và duy trì kiến thức và kỹ năng chuyên môn cần thiết để hoàn thành công việc.
* Thời Gian và Quản Lý Công Việc: Tuân thủ thời gian và quản lý công việc một cách hiệu quả để đảm bảo hoàn thành nhiệm vụ đúng hạn.

## **6.3 Kế hoạch cho kiến thức mới và chiến lược học tập**

Bước 1: Xác định Mục tiêu Học tập

- Xác định rõ mục tiêu học tập của bạn. Bạn muốn học về chủ đề cụ thể nào? Bạn muốn đạt được điều gì sau khi hoàn thành quá trình học?

Bước 2: Tạo Lịch Học Tập

- Xây dựng một lịch học tập cụ thể. Xác định thời gian hàng ngày hoặc hàng tuần bạn dành cho học. Điều này giúp bạn tổ chức thời gian và duy trì kỷ luật học tập.

Bước 3: Xác định Nguồn Tài Liệu

- Tìm các nguồn tài liệu phù hợp với mục tiêu học tập của bạn. Các nguồn tài liệu có thể bao gồm sách, khóa học trực tuyến, bài giảng trực tuyến, video hướng dẫn, blog, và tài liệu tham khảo.

Bước 4: Tạo Kế Hoạch Học Tập

- Xây dựng một kế hoạch học tập cụ thể. Liệt kê ra những chủ đề hoặc phần kiến thức cần học và sắp xếp chúng theo trình tự hợp lý. Điều này giúp bạn tiến triển một cách có hệ thống.

Bước 5: Sử dụng Công Cụ Học Tập

- Sử dụng các công cụ học tập và ghi chép để giúp bạn nắm vững kiến thức. Điều này có thể bao gồm việc tạo ghi chép, tạo sơ đồ tư duy, hoặc sử dụng ứng dụng học tập trực tuyến.

Bước 6: Thiết lập Mục Tiêu Ngắn Hạn và Dài Hạn

- Xác định các mục tiêu ngắn hạn và dài hạn trong quá trình học. Mục tiêu ngắn hạn giúp bạn theo dõi tiến trình hàng ngày hoặc hàng tuần, trong khi mục tiêu dài hạn giúp bạn tập trung vào những kết quả lớn hơn.

Bước 7: Tìm Kiếm Hỗ Trợ và Tương Tác Với Người Khác

- Nếu gặp khó khăn trong quá trình học, đừng ngần ngại tìm kiếm sự hỗ trợ từ giảng viên, người bạn học, hoặc cộng đồng trực tuyến. Tương tác với người khác có thể giúp giải quyết các vấn đề và tìm kiếm câu trả lời cho các câu hỏi của bạn.

Bước 8: Đánh giá và Điều Chỉnh

- Thường xuyên đánh giá tiến trình học tập của bạn và điều chỉnh kế hoạch nếu cần. Điều này giúp bạn duy trì sự linh hoạt và hiệu suất học tập.

Bước 9: Thực Hành và Ứng Dụng Kiến Thức

- Sau khi học kiến thức mới, quan trọng là thực hành và áp dụng nó vào thực tế. Thực hành giúp bạn củng cố kiến thức và phát triển kỹ năng thực tiễn.

# 7. Chương trình Demo

## **7.1. Gameplay**

- Bắt dầu người chơi được nhảy xuống thảm cỏ.

- Mục tiêu chính: nhảy qua các chướng ngại vật để đến đích một cách nhanh chóng.

- Người chơi ấn 2 nút điều hướng trái phải trên bàn phím để di chuyển, phím Space để nhảy lên.

- Khi gặp vật cản nếu người chơi chạm vào nó, lật tức người chơi sẽ bị thua và quay lại bắt đầu của Map để chơi lại từ đầu.

- Người chơi phải qua hết tất cả các màn để dành chiến thắng.

## **7.2. Các chức năng cơ bản**

- Các chức năng chính được đưa ra và test trong demo:

  + Chức năng hiển thị map ( ánh sáng, vật cản bối cảnh)

+ Chức năng nhạc nền

+ Chức năng Player ( di chuyển, nhảy, …)

-Từ góc nhìn thứ 3 người chơi thấy nhân vật nguyên hình và sử dụng không gian tối đa của màn hình để quan sát Map vượt chướng ngại vật.

- Hiện tại trong chương trình demo, nhân vật có đầy đủ chức năng hoạt động.

# 8. Kết luận

## **8.1 Kết quả**

Xây dựng được một trò chơi giải trí không chỉ là tạo ra nó mà còn là cả một quá trình khảo sát thực tế thị hiếu người dùng và những trò chơi hấp dẫn hơn những cái đã được tạo ra bởi những nhà sáng lập trước đó. Nó đòi hỏi chúng ta phải làm bằng cái tâm và thật sự tạo ra được một cái gì đó khác và lạ hơn so với những gì đã có. Em cũng đã tự thực hiện được đam mê của chính mình, tự tạo ra một trò chơi theo phong cách mà mình yêu thích. Tạo ra một trò chơi không chỉ đơn giản là tạo ra một sự giải trí đơn thuần mà còn là gửi gắm vào đó những bài học bổ ích, tính nhân văn và văn hóa.

Đứng trước xu thế phát triển của công nghệ thông tin hiện nay có rất nhiều phần mềm để làm game như: Unity, Unreal Engine, Godot, CryEngine, Lumberyard, GameMaker Studio nhưng em đã thực hiện công việc này trên UNITY 2D . Qua việc nghiên cứu xâydựng game này, chúng em đã phần nào củng cố được các kiến thức về lập trình, cách thiết kế giao diện một cách hài hòa và nắm được cách làm thế nào để xây dựng được một game từ những việc nhỏ nhất như viết cốt truyện cho trò chơi , vẽ giao diện … cho đến những thứ lớn hơn là viết code cho game hoạt động.



## **8.2 Đánh giá**

### 8.2.1 Những điểm làm được

- Xây dựng được Menu: Play, Level, Options, Quit.

- Xây dựng được trang giao diện Paused: Hiển thị được các nút Resume, Restart, Home.

- Chức năng khi hoàn thành 1 màn chơi.

- Xây dựng được giao diện khi thắng.

- Mỗi màn sẽ có 1 cửa để người chơi qua được màn tiếp theo.

- Xây dựng được nhiều chướng ngại vật

### 8.2.2 Những điểm chưa làm được

- Chưa có nhiều hoạt ảnh cho nhân vật

- Chưa có danh sách nhân vật cho người chơi lựa chọn.

- Chưa có nhiều vũ khí cho người chơi lựa chọn.

- Nhân vật di chuyển và gây sát thương lên quái.

- Chưa có chức năng nhặt đồ.

- Chưa có cửa hàng mua đồ.

- Chưa có lưu điểm cao nhất của người chơi.

- Chưa xây dựng được cơ sở dữ liệu.

- Chưa có quái vật tấn công

- Chưa có trụ có khả năng phát hiện nhân vật và tấn công khi nhân vật vào tầm bắn của trụ.

- Chưa có boss cuối

## **8.3 Khó khăn và Học hỏi**

Trong quá trình phát triển game này, em đã đối mặt với nhiều khó khăn và thách thức. Một trong những khía cạnh khó khăn nhất của dự án là quản lý thời gian, tài nguyên và học hỏi những kiến thức mới. Em đã phải làm việc chặt chẽ để đảm bảo tiến độ dự án được duy trì và không bị trễ hạn. Việc phải hoàn thành dự án một mình là một thách thức đáng kể, không những phải sắp xếp thời gian hợp lí mà còn phải học hỏi thêm nhiều kiến thức chuyên sâu.

Không chỉ vậy, em còn đối mặt với các khó khăn kỹ thuật trong việc phát triển game. Trong quá trình này, em đã học cách xử lý các vấn đề liên quan đến tối ưu hóa hiệu suất, đồ họa và âm thanh. Những thách thức này đã giúp em nâng cao kiến thức và kỹ năng kỹ thuật của mình, và chúng ta đã chứng tỏ rằng thách thức là cơ hội để phát triển.

## **8.4 Hướng phát triển**

Do đây là game đầu tay của nhóm nên các chức năng sẻ còn hạn chế. Nhóm sẽ làm tiếp các thiết kế khác như:

- Các màn chơi nâng cao kĩ năng và độ khó cho bớt nhàm chán bởi các màn game mà người chơi đã vượt qua.

- Update nhiều hiệu ứng va chạm, AI cho cácobjects để thêm phần sinh động cũng như trải nghiệm gần nhất với đời thật.

- Thiết kế thêm và hoàn thiện về player (vũ khí, các nâng cấp và các kỹ năng khác).

- Thêm các hiệu năng như thanh máu, mức độ sát thương khi tấn công, mức độ phòng thủ, chạy nhanh, dịch chuyển.

- Thiết kế thêm và thêm chức năng cho Enemy (phù hợp với bản đồ) để cho bản đồ thêm sinh động cũng như thêm độ khó.

- Chức năng unlock vũ khí và cập nhật nhiều vũ khí để game bớt nhàm chán và tạo nên sự mới mẻ.

- Các chức năng điều chỉnh options (âm thanh, ánh sáng, hiệu ứng).

## **8.5 Tổng kết**

Trong tổng kết, dự án làm game này đã mang lại những kết quả quan trọng và giá trị. Em đã thành công trong việc tạo ra một trò chơi thú vị và giáo dục với mục tiêu chính là tạo ra trải nghiệm tương tác hấp dẫn cho người chơi. Sự phản hồi tích cực từ người chơi đã làm cho em tự hào và khích lệ tiếp tục phát triển dự án này trong tương lai.

Dự án này cũng đã giúp em học hỏi nhiều điều quý báu. Em đã học cách làm việc hiệu quả, quản lý dự án một cách chặt chẽ, và nâng cao kỹ năng kỹ thuật của mình. Các thử thách và khó khăn đã giúp em trưởng thành và phát triển sự kiên nhẫn và sức mạnh trong việc giải quyết vấn đề.

Tổng cộng, dự án làm game này không chỉ là một kết quả đáng kể trong việc phát triển game, mà còn là một hành trình học hỏi và trải nghiệm đáng nhớ. Em hy vọng rằng sản phẩm của mình sẽ tiếp tục phát triển và mang lại giá trị cho cộng đồng người chơi.

# LỜI CẢM ƠN

Trong suốt thời gian từ khi bắt đầu học tập tại trường đến nay, em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý Thầy Cô, gia đình và bạn bè. Với lòng biết ơn sâu sắc:

Em chân thành cảm ơn quý thầy cô trường Đại Học Phenikaa đã dạy dỗ, dìu dắt, truyền đạt nhiều kiến thức, kinh nghiệm quý báu.

Đặc biệt với lòng biết ơn sâu sắc nhất em xin gửi lời cảm ơn Thầy Trịnh Thanh Bình là giảng viên trực tiếp hướng dẫn giúp đỡ chúng em trong quá trình thực hiện đồ án chuyên ngành Phân Tích Và Kỹ Thuật Phần Mềm.

Trong thời gian nghiên cứu cũng như trong quá trình làm báo cáo đề tài, không tránh khỏi thiếu sót. Em rất mong nhận được sự góp ý quý báu của quý Thầy Cô, anh chị và các bạn.

Xin chân thành cảm ơn!!!

# TÀI LIỆU THAM KHẢO

<https://learn.unity.com/>

<https://docs.unity.com/>

<https://bom.so/pP18cI>

<https://sd.blackball.lv/library/Unity_in_Action_3rd_Edition_(2022).pdf>