**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**

**VIỆN KỸ THUẬT CÔNG NGHỆ**



**BÁO CÁO TIỂU LUẬN MÔN HỌC**

**TRÍ TUỆ NHÂN TẠO**

**ĐỀ TÀI: XÂY DỰNG BOT CHƠI PACMAN SỬ DỤNG THUẬT TOÁN A\***

**GVHD: ThS.Hoàng Mạnh Hà**

**SVTH: Doãn Quốc Hiếu**

**MSSV: 2124802010547**

**LỚP: D21CNTT02**

**BÌNH DƯƠNG - 02/2024**

**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**

**VIỆN KỸ THUẬT CÔNG NGHỆ**



**BÁO CÁO TIỂU LUẬN MÔN HỌC**

**TRÍ TUỆ NHÂN TẠO**

**ĐỀ TÀI: XÂY DỰNG BOT CHƠI PACMAN SỬ DỤNG THUẬT TOÁN A\***

**GVHD: ThS.Hoàng Mạnh Hà**

**SVTH: Doãn Quốc Hiếu**

**MSSV: 2124802010547**

**LỚP: D21CNTT02**

**BÌNH DƯƠNG - 02/2024**

MỤC LỤC

[MỤC LỤC i](#_Toc160032035)

[DANH MỤC HÌNH iii](#_Toc160032036)

[MỞ ĐẦU 1](#_Toc160032037)

[CHƯƠNG 1. KHẢO SÁT VÀ PHÂN TÍCH BÀI TOÁN 2](#_Toc160032038)

[1.1. Thuật toán A\* 2](#_Toc160032039)

[1.1.1. Giới thiệu thuật toán 2](#_Toc160032040)

[1.1.2. Mô tả thuật toán A\* 2](#_Toc160032041)

[CHƯƠNG 2. GIỚI THIỆU CÔNG NGHỆ SỬ DỤNG 4](#_Toc160032042)

[2.1. Công nghệ sử dụng 4](#_Toc160032043)

[2.1.1. Giới thiệu về Unity 4](#_Toc160032044)

[2.2. Giới thiệu về Visual Studio 6](#_Toc160032046)

[2.3. Ngôn ngữ về C# 7](#_Toc160032047)

[2.4. Ưu điểm của C# 7](#_Toc160032048)

[CHƯƠNG 3. THIẾT KẾ CHƯƠNG TRÌNH 9](#_Toc160032049)

[3.1. Giao diện bắt đầu trò chơi 9](#_Toc160032050)

[3.2. Giao diện PacMan di chuyển trong trò chơi 9](#_Toc160032052)

[3.3. Giao diện khi người dùng sử dụng hết 3 mạng 10](#_Toc160032053)

[KẾT LUẬN 11](#_Toc160032054)

[TÀI LIỆU THAM KHẢO 12](#_Toc160032055)

DANH MỤC HÌNH

[**Hình 2.1:** Unity 4](#_Toc160032142)

[**Hình 2.2:** Visual Studio 6](#_Toc160032144)

[**Hình 3.1:** Giao diện bắt đầu chơi game 9](#_Toc160032145)

[**Hình 3.2:** Giao diện PacMan di chuyển trong trò chơi 10](#_Toc160032147)

[**Hình 3.3:** Giao diện khi người dùng sử dụng hết 3 mạng 10](#_Toc160032149)

MỞ ĐẦU

Trong thế giới nghiên cứu về trò chơi điện tử, trò chơi PacMan đã trở thành một chủ đề hấp dẫn và đa chiều thu hút sự quan tâm của nhiều nhà nghiên cứu. Mục tiêu nghiên cứu của đề tài là khám phá và phát triển các phương pháp và chiến lược để cải thiện hiệu suất chơi của PacMan thông qua việc áp dụng các thuật toán như A\*. Cách tiếp cận trong nghiên cứu này thường xoay quanh việc phân tích cơ chế hoạt động của PacMan và các yếu tố tác động đến quá trình chơi. Việc nghiên cứu thường tiếp cận vấn đề thông qua việc phát triển và thử nghiệm các thuật toán và chiến lược điều khiển sử dụng thuật toán A\* để đưa ra các giải pháp tối ưu cho PacMan, bao gồm việc sử dụng mô phỏng và thử nghiệm trên các môi trường thực tế hoặc ảo để đánh giá hiệu suất của các chiến lược và thuật toán. Đối tượng của nghiên cứu bao gồm cả việc nghiên cứu về cách mà PacMan tương tác với môi trường và cách mà người chơi tương tác với trò chơi. Phạm vi nghiên cứu có thể mở rộng từ việc tối ưu hóa chiến lược di chuyển của PacMan đến việc xử lý các yếu tố phức tạp như sự biến đổi của môi trường, tương tác với các nhân vật khác, hoặc thậm chí là việc áp dụng học máy để cải thiện kỹ năng chơi game của PacMan theo thời gian. Ngoài ra việc nghiên cứu việc áp dụng thuật toán A\* vào trò chơi PacMan không chỉ giúp em hiểu sâu hơn về cơ chế hoạt động của trò chơi mà còn mở ra nhiều cơ hộ để cải thiện và phát triển trí tuệ nhân tạo trong lĩnh vực trò chơi điện tử. Tuy nhiên, sản phẩm này của em về chuyên môn vẫn chưa được hoàn thiện và còn một số thiếu sót. Nhưng em sẽ cố gắng hoàn thiện ứng dụng một cách chỉnh chu và chuyên nghiệp nhất có thể .

1. KHẢO SÁT VÀ PHÂN TÍCH BÀI TOÁN
   1. Thuật toán A\*
      1. Giới thiệu thuật toán

Thuật toán A\* được mô tả lần đầu tiên năm 1986 bởi Peter Hart, Nils Nilson và Bertram Raphael. Trong báo cáo của họ, thuật toán được gọi là thuật toán A, khi sử dụng thuật toán này với một hàm đánh giá heuristic thích hợp sẽ thu được hoạt động tối ưu, do đó mà có tên là A\*.

Trong khoa học máy tính, A\* là một thuật toán tìm kiếm trong đồ thị. Thuật toán này tìm một đường đi từ nút khởi đầu tới một nút đích cho trước (hoặc tới một nút thỏa mãn điều kiện đích). Thuật toán này sử dụng một đánh giá heuristic để xếp loại từng nút theo ước lượng về tuyến đường tốt nhất đến nút đó. Thuật toán này duyệt các nút theo thứ tự của đánh giá heuristic này. Do đó, thuật toán A\* là một ví dụ của tìm kiếm theo lựa chọn tốt nhất (best-first search).

Xét bài toán tìm đường – bài toán mà A\* thường được dùng để giải. A\* xây  
dựng tăng dần tất cả các tuyến đường từ điểm xuất phát cho tới khi nó thấy một đường đi chạm tới đích. Tuy nhiên, cũng như tất cả các thuật toán tìm kiếm có thông tin nó chỉ xây dựng các tuyến đường có vẻ dần về đích.

Để biết những tuyến đường nào có khả năng sẽ dẫn tới đích, A\* sử dụng một  
hàm đánh giá heuristic về khoảng cách từ điểm bất kỳ cho tới đích. Trong  
trường hợp tìm đường đi, đánh giá này có thể là khoảng cách đường chim bay - một đánh giá xấy xỉ thường dùng cho khoảng cách của đường giao thông.

Điểm khác biệt của A\* đối với tìm kiếm theo lựa chọn tốt nhất là nó tính đến  
khoảng cách đã đi qua. Điều đó làm cho A\* đầy đủ và tối ưu, nghĩa là A\* sẽ  
luôn tìm thấy đường đi ngắn nhất nếu tồn tại một đường đi như thế. A\* không  
đảm bảo sẽ chạy nhanh hơn các thuật toán tìm kiếm đơn giản hơn. Trong một môi trường dạng mê cung, cách duy nhất để đến đích có thể là trước hết phải đi về phía xa đích và cuối cùng mới quay trở lại. Trong trường hợp đó, việc thử các nút theo thứ tự “gấn đích hơn thì được thử trước” có thể gây tốn thời gian.

* + 1. Mô tả thuật toán A\*

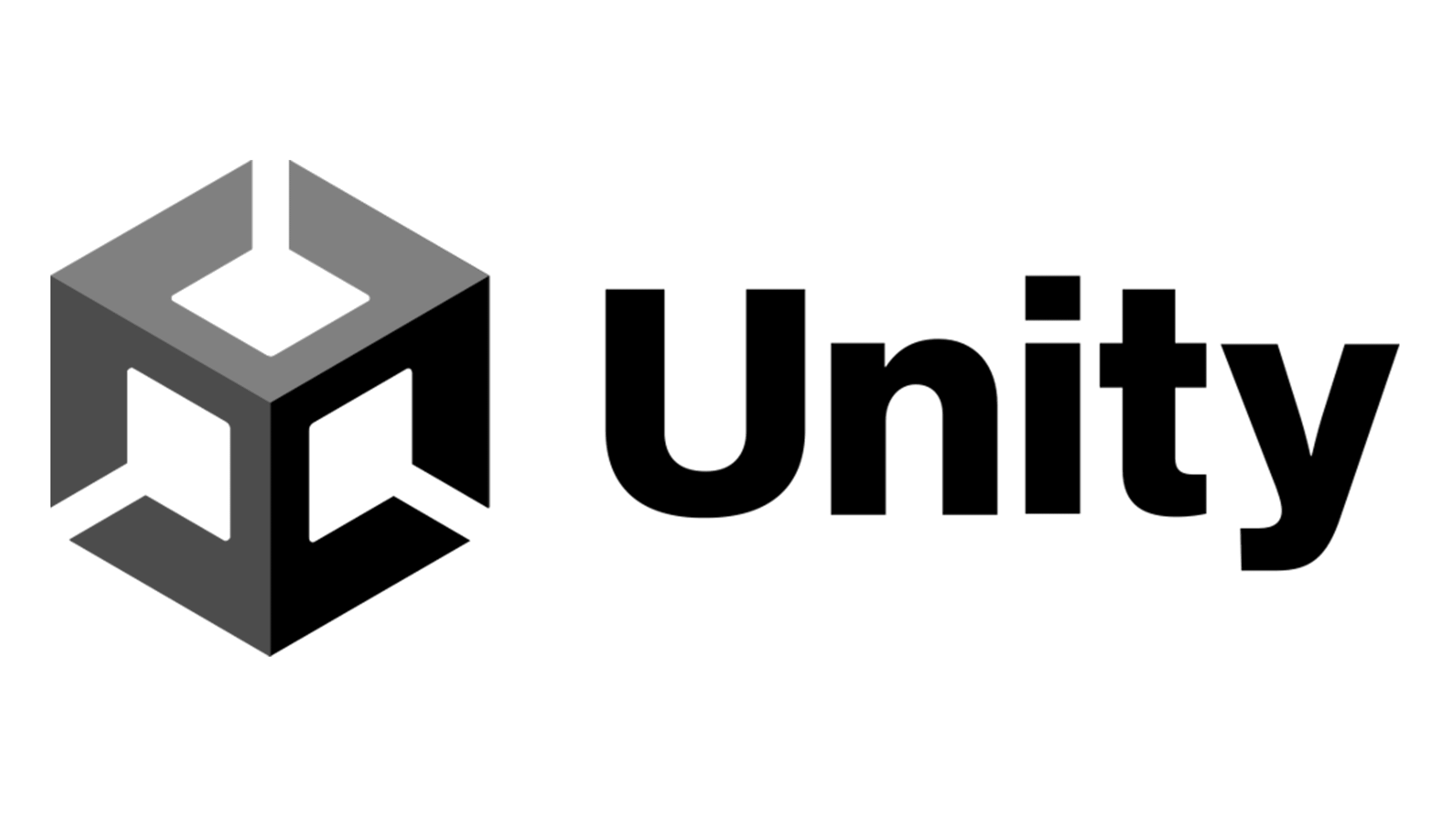
Giả sử n là một trạng thái đạt tới (có đường đi từ trạng thái ban đầu n0 tới n). Ta xác định hàm đánh giá: f(n)=g(n)+h(n)

* g(n) là chi phí từ nút gốc 10 tới nút hiện tại n
* h(n) chi phí ước lượng từ nút hiện tại n tới đích
* f(n) chi phí tổng thể ước lượng của đường đi qua nút hiện tại a đến đích

Một ước lượng heuristic h(n) được xem là chấp nhận được nếu với mọi  
nút n: 0< h(n)<h\*(n)

Trong đó h\*(n) là chi phí thật (thực tế) để đi từ nút n đến đích.

1. GIỚI THIỆU CÔNG NGHỆ SỬ DỤNG
   1. Công nghệ sử dụng
      1. Giới thiệu về Unity



* + - 1. Unity

Unity hiểu một cách đơn giản nhất chính là phần mềm được sử dụng để làm game đa nền tảng và được phát triển bởi Unity Technologies. Phần mềm này chủ yếu áp dụng cho hệ điều hành trên máy tính, điện thoại và consoles, được công bố lần đầu tiên trên thế giới với hệ điều hành của OS X vào năm 2005. Đến nay, phần mềm này đã được phát triển, mở rộng ra đến 27 nền tảng mới.

Phần mềm phát triển game unity hỗ trợ cho người dùng đầy đủ đồ họa 2D và 3D cùng rất nhiều chức năng, chủ yếu được viết nên bởi ngôn ngữ lập trình C#. Ngoài ra, 2 ngôn ngữ khác cũng được sử dụng để hỗ trợ đó là Boo và Unityscript, tuy nhiên đều đã bị loại ở unity 5 và unity 2017. Unity cũng hỗ trợ cho người dùng APIs ở cấp thấp như là Metal trên hệ điều hành IOS, macOS và Vulkan trên hệ điều hành Android. Hiện nay, unity là phần mềm đang được sử dụng phổ biến nhất tại các tổ chức, doanh nghiệp chuyên về phát triển game trên thế giới cũng như tại Việt Nam, góp phần mang đến những trò game cùng các trải nghiệm vô cùng thú vị.

Unity luôn là sự lựa chọn tuyệt vời dành cho các lập trình viên chuyên xây dựng và phát triển mảng game, bởi nó sở hữu vô số ưu điểm nổi bật như sau:

* Có đa dạng các công cụ hỗ trợ như vật lý, âm thanh, mã nguồn, trí tuệ nhân tạo... và bao gồm cả việc dựng hình ảnh để cho ra đồ họa 2D/3D bắt mắt.
* Với Unity, người dùng có thể dễ dàng tạo ra dòng dữ liệu để xử lý và việc kết nối  
  hệ thống mạng lưới cũng trở nên đơn giản hơn...
* Quá trình xây dựng, phát triển và lập trình game vô cùng đơn giản. Đồng thời, giúp tiết kiệm thời gian trong mọi mặt.
* Hỗ trợ đa nền tảng như iOS, Android, Windows, PlayStation 3, Xbox 360, Linux...
* Nhà phát triển chỉ cần một gói lập trình duy nhất là đã có thể phát triển game phục vụ được trên nhiều hệ điều hành.
* Lập trình viên cũng dễ dàng convert sang các hệ điều hành khác nhau, tùy theo mục đích và yêu cầu của người đặt.
* Unity là phần mềm rất dễ sử dụng, nó được đánh giá như môi trường tích hợp và mang đến cho người dùng một hệ thống toàn diện.
* Lập trình viên cũng có thể soạn thảo mã nguồn, xây dựng công cụ tự động hóa riêng và sửa lỗi chúng... một cách dễ dàng, nhanh chóng.
* Cộng đồng người sử dụng lập trình Unity khá đông. Chính vì vậy mà trong quá trình làm việc, nếu bạn gặp bất kỳ vấn đề nào thì đều sẽ nhanh chóng nhận được sự hỗ trợ từ những người có kinh nghiệm lâu năm liên quan đến công nghệ phát triển game.
* Những người lựa chọn lập trình Unity để phát triển sự nghiệp sẽ nhận được mức thu nhập tương đối cao, có thể lên đến 1500USD/năm.
* Trong Unity có rất nhiều công cụ nhỏ khác giúp phục vụ cho các mục đích khác nhau trong quá trình lập trình viên code game. Cụ thể, dưới đây là một số khái niệm thường được sử dụng trong quá trình phát triển game bằng Unity Engine:
* **GameObject**: Đây là một đối tượng cụ thể trong game như là nhân vật hay đồ vật nào đó (con người, xe cộ, động vật,…)
* **Component**: Đối với mỗi GameObject đều sẽ có nhiều thành phần khác nhau cấu tạo nên như là hình ảnh, các hành động, tính toán vật lý, thành phần để xử lí va chạm, các mã điều khiển,... và mỗi thành phần này sẽ được gọi là component.
* **Sprite**: đây là component đại diện cho một hình ảnh 2D của game.
* **Animation**: đây là tập hợp các hình ảnh thay đổi liên tục qua đó tạo thành một hoạt ảnh.
* **Prefabs**: khái niệm này được dùng để chỉ việc sử dụng lại các đối tượng giống nhau trong game chỉ cần thực hiện thao tác khởi tạo lại các giá trị của vị trí cùng các tỉ lệ biến dạng, góc quay từ đối tượng ban đầu.
* **Script**: đây là tập tin có chứa đựng những đoạn mã nguồn và các lập trình viên game sẽ sử dụng nó để khởi tạo, xử lý các đối tượng ở trong game. Và trong unity hiện nay có thể sử dụng các ngôn ngữ như là Java Script, C# hay Boo để lập trình Script.
* **Scenes**: yếu tố quan trọng để quản lý tất cả những đối tượng có trong một trận game.
* **Assets**: bao gồm toàn bộ những công cụ khác phục vụ cho một dự án game đó là  
  sprite, sound, script, animation, scenes,...
* **Camera**: đây là một gameObject đặc biệt có trong scenes và được sử dụng để xác  
  định tầm nhìn cũng như quan sát các đối tượng khác có trong game.
* **Transform**: Hiểu đơn giản đây là 3 phép biến đổi trong game bao gồm có tịnh tiến, phóng to – thu nhỏ và quay theo các trục của một đối tượng nhất định
  1. Giới thiệu về Visual Studio



* + - 1. Visual Studio

Visual Studio chính là một môi trường tích hợp được phát triển bởi Microsoft, ứng dụng được sử dụng để phát triển các chương trình máy tính bao gồm các trang web, ứng dụng web, dịch vụ web và ứng dụng di động. Visual Studio sử dụng trên các nền tảng phát triển phần mềm của Microsoft như Windows API, Windows Forms, Windows Presentation Foundation, Windows Store và Microsoft Silverlight. Nó có thể tạo ra cả mã gốc và mã được quản lý. Visual Studio hỗ trợ 36 ngôn ngữ lập trình khác nhau và cho phép trình soạn thảo mã và trình gỡ lỗi hỗ trợ (ở các mức độ khác nhau) gần như bất kỳ ngôn ngữ lập trình nào, miễn là dịch vụ ngôn ngữ cụ thể tồn tại. Các ngôn ngữ tích hợp sẵn bao gồm C,[6] C++, C++/CLI, Visual Basic NET, C#, F#,[7] JavaScript, TypeScript, XML, XSLT, HTML, và CSS. Hỗ trợ cho các ngôn ngữ khác như Python,ll Ruby, Node.js và M trong số những ngôn ngữ khác có sẵn thông qua các trình cắm. Java (và J #) đã được hỗ trợ trong quá khứ. Trình soạn thảo này vận hành mượt mà trên các nền tảng như Windows, macOS, Linux.

Phiên bản cơ bản nhất của Visual Studio, phiên bản Cộng đồng, có sẵn miễn phí. Khẩu hiệu cho phiên bản Visual Studio Community là "IDE miễn phí, đầy đủ tính năng cho sinh viên, nhà phát triển mã nguồn mở và cá nhân". Tính đến ngày 19 tháng 2 năm 2024, Visual Studio 2022 là phiên bản sẵn sàng sản xuất hiện tại. Visual Studio 2013, 2015 và 2017 đang được hỗ trợ mở rộng, trong khi Hỗ trợ Chính năm 2019

* 1. Ngôn ngữ về C#

C# (hay C sharp) là một ngôn ngữ lập trình đơn giản, được phát triển bởi đội ngũ kỹ sư của Microsoft vào năm 2000. C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.

Trong các ứng dụng Windows truyền thống, mã nguồn chương trình được biên dịch trực tiếp thành mã thực thi của hệ điều hành. Trong các ứng dụng sử dụng .NET Framework, mã nguồn chương trình (C#, VB.NET) được biên dịch thành mã ngôn ngữ trung gian MSIL (Microsoft intermediate language).

Sau đó mã này được biên dịch bởi Common Language Runtime (CLR) để trở thành  
mã thực thi của hệ điều hành. Hình bên dưới thể hiện quá trình chuyển đổi MSIL code  
thành native code.

* 1. Ưu điểm của C#

C# là ngôn ngữ đơn giản: C# loại bỏ một vài sự phức tạp và rối rắm của những ngôn ngữ như Java và C++ bao gồm việc loại bỏ những marco, những đa kế thừa, và lớp cơ sở ảo (virtual base class).

Ngôn ngữ C# đơn giản vì nó dựa trên nền tảng C và C++. Nếu chúng ta thân  
thiện với C và C++ hoặc thậm chí là Java, chúng ta sẽ thấy C# khá giống về diện mạo, cú pháp, biểu thức, toán tử và những chức năng khác được lấy trực tiếp từ ngôn ngữ C và C++, nhưng nó đã được cải tiến để làm cho ngôn ngữ đơn giản hơn.

C# là ngôn ngữ hiện đại: Điều gì làm cho một ngôn ngữ hiện đại? Những đặc tính như là xử lý ngoại lệ, thu gom bộ nhớ tự động, những kiểu dữ liệu mở rộng, và bảo mật mã nguồn là những đặc tính được mong đợi trong một ngôn ngữ hiện đại. C# chứa tất cả những đặc tính trên. Nếu là người mới học lập trình có thể chúng ta sẽ cảm thấy những đặc tính trên phức tạp và khó hiểu. Tuy nhiên, cũng đừng lo lắng chúng ta sẽ dần dần được tìm hiểu những đặc tính qua các nội dung khóa học này.

C# là một ngôn ngữ lập trình thuần hướng đối tượng: Lập trình hướng đối tượng (OOP: Object-oriented programming) là một phương pháp lập trình có 4 tính chất. Đó là tính trừu tượng (abstraction), tính đóng gói (encapsulation), tính đa hình (polymorphism) và tính kế thừa (inheritance). C# hỗ trợ cho chúng ta tất cả những đặc tính trên.

C# là một ngôn ngữ ít từ khóa : C# là ngôn ngữ sử dụng giới hạn những từ khóa. Phần lớn các từ khóa được sử dụng để mô tả thông tin. Chúng ta có thể nghĩ rằng một ngôn ngữ có nhiều từ khóa thì sẽ mạnh hơn. Điều này không phải sự thật, ít nhất là trong trường hợp ngôn ngữ C#, chúng ta có thể tìm thấy rằng ngôn ngữ này có thể được sử dụng để làm bất cứ nhiệm vụ nào.

1. THIẾT KẾ CHƯƠNG TRÌNH
   1. Giao diện bắt đầu trò chơi

A video game with a square maze

Description automatically generated

* + - 1. Giao diện bắt đầu chơi game

Giao diện này là giao diện ban đầu khi người dùng bắt đầu vào trò chơi, người dùng di chuyển PacMan bằng các phím di chuyển lên,xuống, trái, phải trên bàn phím. Nhằm tránh được các con BOT trong bản đồ và ăn những hạt nhỏ trong bản đồ nhằm tích lũy điểm.

* 1. Giao diện PacMan di chuyển trong trò chơi

A video game with a square maze

Description automatically generated

* + - 1. Giao diện PacMan di chuyển trong trò chơi

Ở giao diện này PacMan đã di chuyển ăn những hạt trong bản đồ nhằm tích lũy điểm thưởng, số điểm thưởng người dùng đạt được hiển thị ở góc trái màn hình

* 1. Giao diện khi người dùng sử dụng hết 3 mạng

A screen shot of a video game

Description automatically generated

* + - 1. Giao diện khi người dùng sử dụng hết 3 mạng

Giao diện này khi người dùng tham gia trò chơi và đã sử dụng hết 3 mạng màn hình sẽ thông báo “GAME OVER”

KẾT LUẬN

**1. Kết quả đạt được**

Sau 4 tuần thực hiện, với nhiều nỗ lực và cũng nhận được nhiều sự giúp đỡ của thầy cô và bạn bè, em thực hiện đề tài đã thực hiện được các chức năng cơ bản như đã đặt ra từ ban đầu của báo cáo. Qua đó, phần nào cũng nâng cao được khả năng lập trình cũng như việc tự nghiên cứu, tìm hiểu thông các tài liệu khác nhau, thực sự giúp ích cho công việc của bản thân sau khi ra trường. Về phần mềm ứng dụng xây dụng BOT chơi PACMAN sử dụng thuật toán A\*, có thể nói chương trình đã có thể vận hành được một cách cơ bản

Về phần người dùng có di chuyển được PacMan trong trò chơi, cơ bản đã hoàn thiện được một vài đặc điểm của trò chơi.

Về phần trò chơi:

* Xây dựng được bản đồ
* Hiển thị điểm thưởng tích lũy
* Hiển thị số mạng còn lại
* Áp dụng cơ bản được thuật toán A\* vào trò chơi

Bên cạnh những kết quả đạt được song song với đó là những việc em vẫn chưa hoàn thiện được trong đề tài lần này:

* Phân hóa được độ khó trong trò chơi
* Xây dựng nhiều bản đồ trong trò chơi
* Trò chơi chỉ đang thực nghiệm ở môi trường lập trình

**2. Hướng phát triển của đề tài**

Thời gian tới em sẽ cố gắng nghiên cứu, và bổ sung một vài chức năng cho trò chơi : Thực tế ảo (VR), xây dựng thành một trò chơi trên các nền tảng di động…..

Giao diện đẹp, thân thiện với người dùng, có nhiều chức năng và tương tác tốt hơn với người dùng.

Thiết kế, bổ sung các tính năng còn thiếu hay chưa thật sự hoàn chỉnh trong quá trình sử dụng.

TÀI LIỆU THAM KHẢO

**Tài liệu website**

[1] <https://docs.unity.com/> - Unity Documentation [Truy cập ngày 25/02/2024]

[2] <https://tailieu.hiepsiit.com/view/bao-cao> – Tài liệu báo [Truy cập ngày 15/02/2024]

[3] <https://assetstore.unity.com/> - Nguồn tài nguyên xây dựng Game [Truy cập ngày 26/02/2024

[4]<https://viblo.asia/p/gioi-thieu-ve-unity-engine-game-engine-pho-bien-nhat-hien-nay-V3m5WBj8lO7> - Tổng quan về Unity [Truy cập ngày 24/02/2024]

[5]<https://gameinternals.com/understanding-pac-man-ghost-behavior> -UnderstandPacman [Truy cập ngày 27/02/2024]