

Course - Lập trình AI cho Unreal Engine

---

# Giới thiệu Game AI

---

*Chương này sẽ giới thiệu ý tưởng cơ bản về Trí tuệ nhân tạo (AI) và cách nó ảnh hưởng trực tiếp cũng như nâng cao trải nghiệm người dùng khi chơi trò chơi.*

Tags: unreal engine, game ai, trí tuệ nhân tạo cho game

## **Ai nên đọc tài liệu này**

Tài liệu này dành cho các lập trình viên và nghệ sĩ muốn mở rộng kiến thức về AI trong trò chơi liên quan đến Unreal Engine 4. Bạn nên có một số kinh nghiệm khám phá Unreal Engine 4 trước khi tìm hiểu tài liệu này vì chúng ta sẽ đi thẳng vào AI trong trò chơi.

## **Giới thiệu sơ quát game AI**

Chương này sẽ giới thiệu ý tưởng cơ bản về Trí tuệ nhân tạo (AI) và cách nó ảnh hưởng trực tiếp cũng như nâng cao trải nghiệm chơi trò chơi. Bạn sẽ tìm hiểu sự khác biệt giữa cách tiếp cận truyền thống và cách tiếp cận trò chơi cụ thể của AI. Chúng tôi sẽ giới thiệu các kỹ thuật khác nhau được sử dụng trong trò chơi AI, bao gồm điều hướng, Behavior Trees, Sensor System (Giác quan), v.v. Bạn sẽ tìm hiểu ngắn gọn những công cụ nào chúng tôi sử dụng cho AI trong editor của Unreal Engine 4. Sau chương này, độc giả sẽ có được hiểu biết cơ bản về cách áp dụng AI vào phát triển trò chơi để có trải nghiệm chơi trò chơi tốt hơn. Các kỹ thuật AI mà chúng tôi sẽ đề cập ngắn gọn ở đây sẽ được giảng dạy trong các chương tiếp theo.

## **Trí tuệ nhân tạo cho game**

Khi bạn lần đầu tiên nghĩ đến Trí tuệ nhân tạo, người máy sẽ ngay lập tức được nghĩ đến. AI bắt nguồn từ ý tưởng về trí thông minh giúp các sinh vật sống đưa ra quyết định. Chúng tôi lấy thông tin đầu vào, bối cảnh và lý do cá nhân của mình để quyết định các hành động mà chúng tôi sẽ thực hiện. Trong AI, chúng tôi cố gắng tái tạo phần ảo của quá trình này để tạo ra các hệ thống có thể có hành vi tự trị. Giả sử bạn có lịch sử chơi game khá dày dặn, bạn sẽ biết rằng AI trong trò chơi nói chung không thông minh hơn một số trò chơi cũ hơn, nơi kẻ thù của bạn có thể bị mắc kẹt trong một góc nào đó và không thể thoát ra được khỏi đó. Game AI bây giờ không thể so sánh với AI nói chung trong nghiên cứu khoa học. Trí tuệ nhân tạo của trò chơi được thiết kế để hoạt động trong một thế giới ảo có thể dự đoán và được kiểm soát tốt. Nó chủ yếu bao gồm các quy tắc được mã hóa cứng để cho phép các actor của trò chơi thực hiện các hành động phù hợp tương ứng với các tình huống khác nhau. Trò chơi AI nhằm mục đích vui nhộn, vì vậy nó chỉ cần có vẻ thông minh đối với người chơi trong bối cảnh này.

Công bằng mà nói, AI là một chủ đề rất rộng, vì vậy việc triển khai mọi kỹ thuật có thể không phải là kế hoạch. Vì vậy, không cần phải nói rằng chúng ta sẽ chỉ đề cập đến những gì cần thiết để bạn tạo ra một trò chơi AI tuyệt vời. Tuy nhiên, hãy nhớ rằng chúng ta sẽ chỉ đề cập đến các kỹ thuật AI trò chơi rất cụ thể; thế giới của AI rộng lớn đến mức tuyệt vời.

## Cách AI của game ảnh hưởng đến trải nghiệm chơi

Người chơi tìm kiếm trải nghiệm chân thực và sống động trong trò chơi. AI đóng một vai trò quan trọng trong việc hình thành trải nghiệm chơi trò chơi này bằng cách mang lại sự chân thực và thú vị cho thế giới ảo. Hãy tưởng tượng rằng bạn đi cùng với một chú chó khi bạn đi dạo xung quanh hoặc một đàn chim chạy tán loạn khi bạn gây ra tiếng động nào đó. Đối thủ của kẻ thù có lẽ là cách triển khai phổ biến và quan trọng nhất của trò chơi AI. Một vài loại trò chơi AI—điều hướng, chiến đấu, hỗ trợ hoặc phân tích—bổ sung các yếu tố còn thiếu cho những người chơi khác để khiến họ cảm thấy cạnh tranh thực sự và đầy thử thách. Điều này bắt nguồn từ khi nó được sử dụng nhiều nhất trong Cờ vua (Chess), Nim, Pong và Pac-Man. Cho đến nay, nó vẫn được sử dụng trong war frame (khung hình chiến sự), với các màn chơi được xây dựng theo thủ tục. Khi thiết kế trò chơi hiện đại phát triển một cách nhanh chóng bằng cách giới thiệu các tính năng mới để chơi trò chơi, chẳng hạn như thế giới mở, nhân vật trong trò chơi khổng lồ và tương tác xã hội, nó cũng gây ra các vấn đề vì các tính năng này khiến việc ra quyết định của AI cần nhiều thông tin đầu vào hơn trong môi trường không thể đoán trước. Ngay cả bây giờ, các tựa game AAA có sự phức tạp của riêng chúng với AI dẫn đến sự hài lòng của người dùng kém đi. Chúng tôi sẽ giới thiệu trong các phần sau một số kỹ thuật mạnh mẽ để giúp tạo mô-đun AI quan trọng này và thảo luận về cách chúng được triển khai trong Unreal Engine.

## Các kỹ thuật và bài tập về Game AI

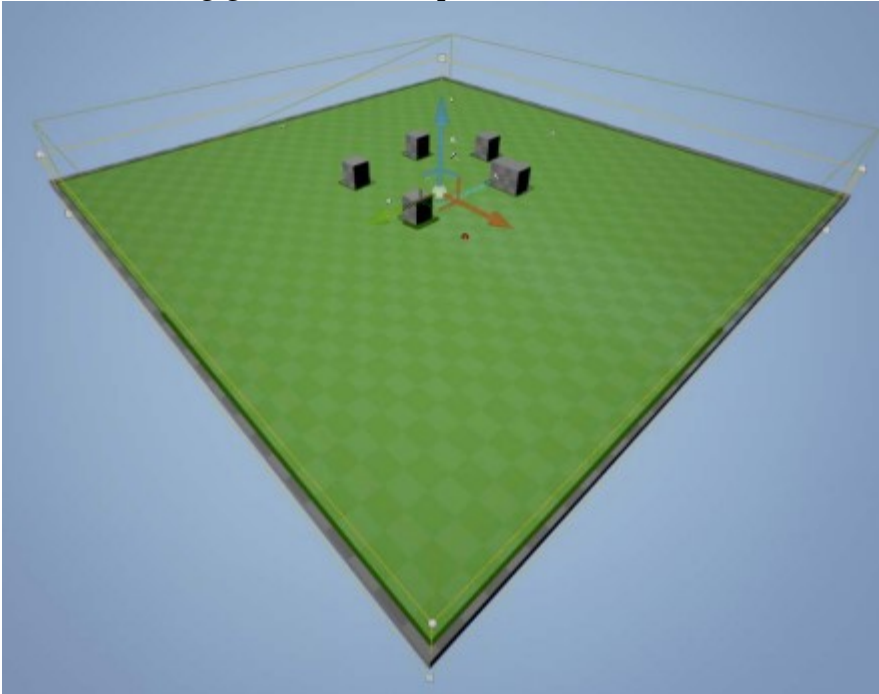
Có nhiều kỹ thuật để bao gồm các khía cạnh khác nhau trong trò chơi AI, từ chuyển động cơ bản đến cảm quan môi trường tiên tiến và ra quyết định. Hãy xem xét từng cái một.

### Sự điều hướng (navigation)

Điều hướng cho AI thường được xây dựng từ các công cụ sau:

- **Navigation Mesh:** Sử dụng các công cụ như Navigation Mesh, còn được gọi là NavMesh, bạn có thể chỉ định các khu vực mà AI có thể đi qua. NavMesh là một biểu diễn đa giác đơn giản hóa của một màn chơi (vùng màu xanh lá cây trong ảnh chụp màn hình bên dưới), trong đó mỗi đa giác hoạt động như một node đơn lẻ được kết nối với các node gần đó. Thông thường, quá trình này được tự động hóa và không yêu cầu các nhà thiết kế đặt các node theo cách thủ công. Sử dụng các công cụ đặc biệt trong Unreal, họ phân tích hình học của màn chơi và tạo Navigation Mesh được tối ưu hóa nhất cho phù hợp. Tất nhiên, mục đích là để xác định các khu vực có thể chơi được trong màn chơi bởi các actor trong trò chơi. Lưu ý rằng đây là kỹ thuật tìm đường duy nhất có sẵn; chúng ta sẽ sử dụng NavMesh trong các ví dụ được cung cấp trong cuốn sách này vì nó hoạt động tốt trong phần trình diễn demo này.

- **Path Following (Path Nodes):** Một giải pháp tương tự cho NavMesh, Path nodes có thể chỉ định không gian mà AI đi qua:



- **Behavior Tree:** Sử dụng Behavior Tree để tác động đến điểm đến tiếp theo của AI có thể tạo ra trải nghiệm người chơi đa dạng hơn. Nó không chỉ tính toán điểm đến được yêu cầu mà còn quyết định xem nó có nên vào màn hình bằng động tác bật lộn ngược hai vòng, không dùng tay hay lộn nhào ba lần và có tay chống đỡ.
- **Steering:** Hành vi bẻ lái (Steering Behavior) có ảnh hưởng đến cách AI di chuyển trong khi điều hướng để tránh chướng ngại vật. Điều này cũng có nghĩa là sử dụng Steering để tạo đội hình với các hạm đội của bạn mà bạn đã thiết lập để tấn công bức tường của nhà vua. Steering có thể được sử dụng theo nhiều cách để tác động đến chuyển động của nhân vật.
- **Sensor System:** Sensory System (Hệ thống cảm quan) có thể cung cấp các chi tiết quan trọng, chẳng hạn như cảm nhận người chơi ở gần đó, mức độ của âm thanh, rào chắn ở gần đó và nhiều biến số khác của môi trường có thể thay đổi chuyển động. Điều quan trọng là trí tuệ nhân tạo của bạn phải hiểu được sự thay đổi của môi trường để nó không nghĩ rằng nó là một anh hùng thực sự.

Mặc dù tất cả các thành phần này không cần thiết để đạt được điều hướng AI, nhưng tất cả chúng đều cung cấp phản hồi quan trọng, có thể ảnh hưởng đến điều hướng. Điều hướng trong một thế giới chỉ bị giới hạn bởi các con đường trong trò chơi. Chúng ta có thể xem một ví dụ về hành vi nhóm với một số thành viên đi theo người lãnh đạo ở đây:



## Chuyển động thực hơn với Steering

Khi bạn nghĩ về chức năng của Steering đối với ô tô, bạn sẽ có lý khi hình dung ý tưởng tương tự được áp dụng cho điều hướng AI trong trò chơi. Steering ảnh hưởng đến chuyển động của AI khi nó đi đến đích tiếp theo. Các ảnh hưởng có thể được cung cấp khi cần thiết, nhưng chúng tôi sẽ xem xét những ảnh hưởng được sử dụng phổ biến nhất. Tránh được sử dụng để tránh va chạm với AI đang tới. Flocking (Chia đàn) là một yếu tố quan trọng khác trong việc lái và rất hữu ích trong việc mô phỏng chuyển động nhóm thú vị, chẳng hạn như tình huống hoảng loạn hoàn toàn hoặc đàn cá. Mục tiêu của các hành vi Steering là đạt được chuyển động và hành vi thực tế trong thế giới của người chơi.

## Tạo một nhân vật với tính ngẫu nhiên và xác suất

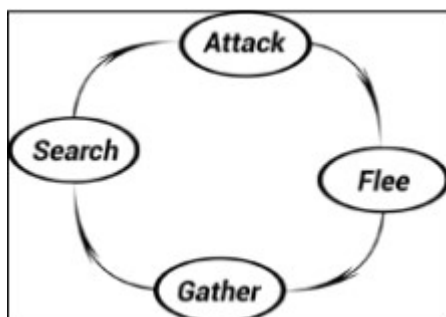
AI với nhân vật là thứ mà tính ngẫu nhiên và xác suất bổ sung vào khả năng ra quyết định của bot. Nếu một bot tấn công bạn theo cùng một cách, luôn vào hiện trường theo cùng một cách và khiến bạn khó chịu vì tiếng cười của nó sau mỗi lần tấn công thành công, thì điều đó sẽ không tạo ra trải nghiệm độc đáo. Thay vào đó, sử dụng tính ngẫu nhiên và xác suất, bạn có thể khiến AI cười dựa trên xác suất hoặc đưa tính ngẫu nhiên vào kỹ năng lựa chọn của AI. Một sản phẩm phụ tuyệt vời khác của việc áp dụng tính ngẫu nhiên và xác suất là nó cho

phép bạn đưa ra các cấp độ khó hoặc giảm khả năng bỏ lỡ kỹ năng sử dụng và thậm chí cho phép các bot nhắm mục tiêu chính xác hơn. Nếu bạn có các bot đi lang thang tìm kiếm kẻ thù, xác suất và tính ngẫu nhiên có thể được sử dụng để làm việc với đầu vào cảm giác của bot nhằm đưa ra quyết định hợp lý hơn.

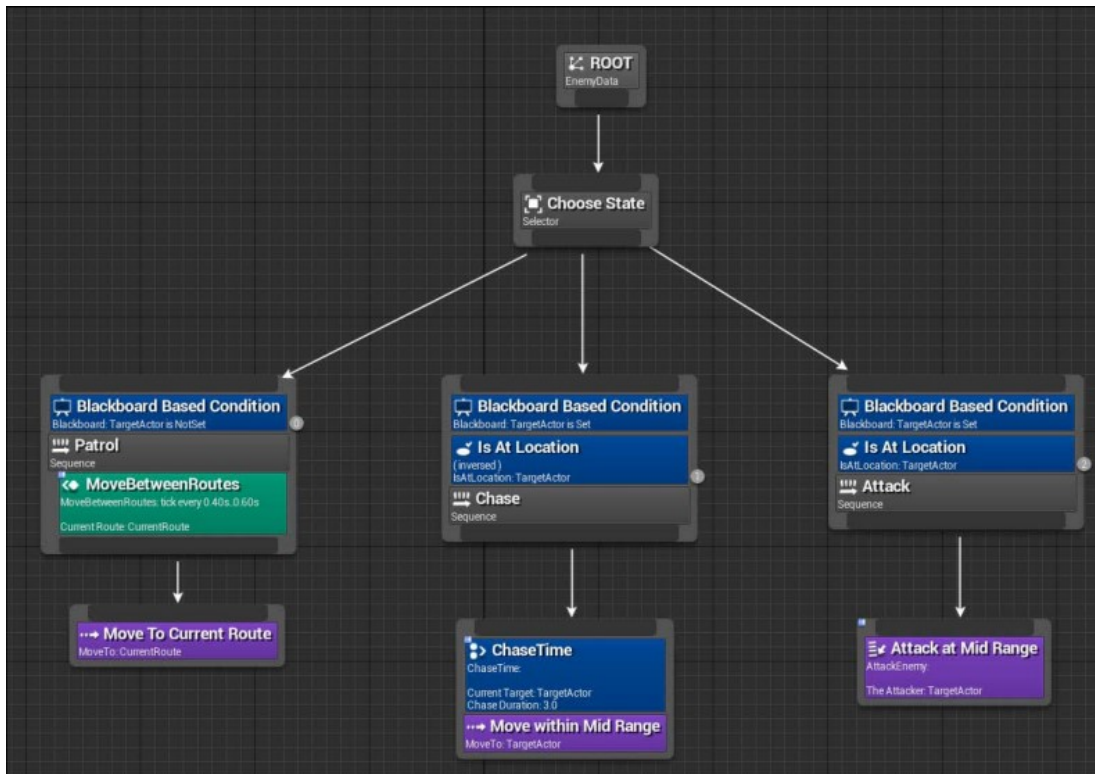
## Tạo ra quyết định phức tạp với Behavior Tree

**Finite State Machines (FSM)** là một mô hình để xác định cách một số trạng thái chuyển tiếp lẫn nhau. Ví dụ, điều này cho phép nó chuyển từ thu thập sang tìm kiếm rồi tấn công, như thể hiện trong hình minh họa dưới đây. Behavior cũng tương tự như vậy, nhưng chúng có phần linh hoạt hơn. Behavior Tree cho phép FSM phân cấp, giới thiệu một số tầng quyết định khác. Vì vậy, bot sẽ quyết định giữa các nhánh hành vi rồi xác định trạng thái của nó. Có một công cụ do UE4 cung cấp có tên là Behavior Tree. Công cụ editor này cho phép chúng ta sửa đổi hành vi của AI một cách nhanh chóng và dễ dàng.

Đây là sơ đồ của mô hình FSM:



Hãy xem qua các components của Behavior Tree



Bây giờ, chúng ta sẽ thảo luận về các components được tìm thấy trong Behavior Tree của UE4.

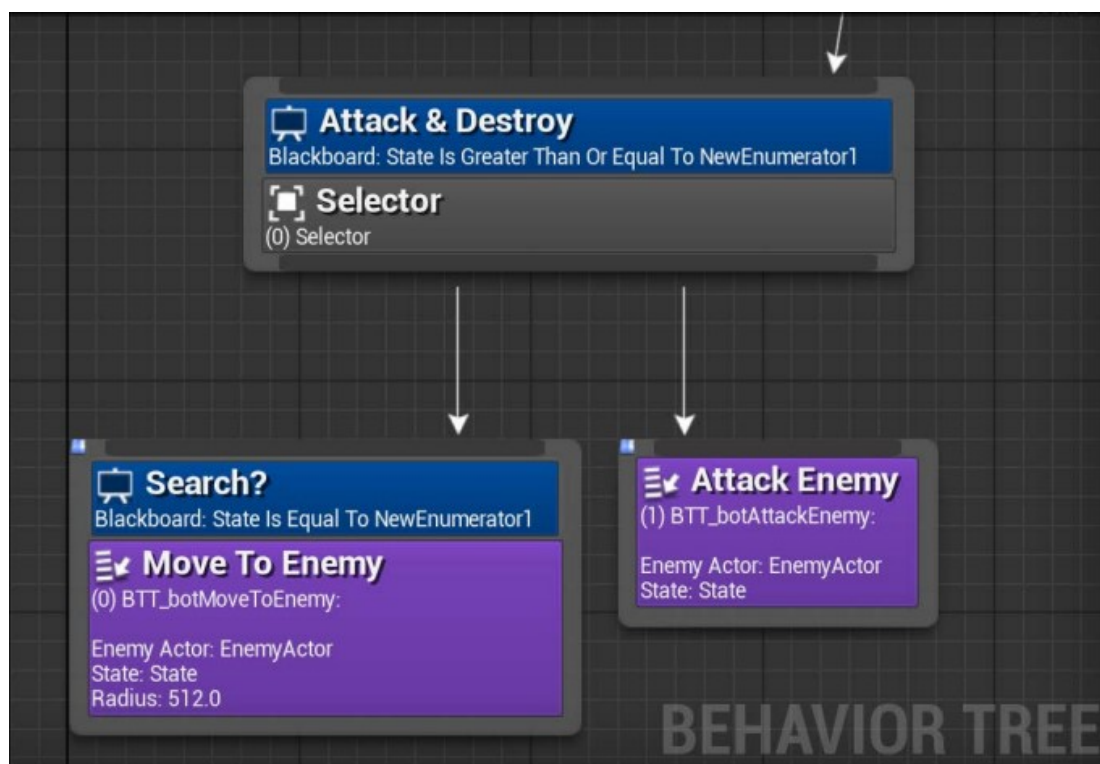
## Root

Node này là node đầu tiên sẽ gửi tín hiệu đến node tiếp theo trong sơ đồ cây. Nó kết nối với một hỗn hợp, thứ mà sẽ bắt đầu cây đầu tiên của bạn. Những gì bạn có thể nhận thấy là trước tiên bạn được yêu cầu sử dụng một hỗn hợp để xác định một cây và sau đó tạo tác vụ cho cây này. Điều này là do FSM phân cấp tạo ra các nhánh của các trạng thái. Các trạng thái này sẽ được phổ biến với các trạng thái hoặc nhiệm vụ khác. Điều này cho phép chuyển đổi dễ dàng giữa nhiều trạng thái. Bạn có thể thấy node gốc trông như thế nào trong ảnh chụp màn hình sau:



## Decorators

Decorators là các câu lệnh có điều kiện (phần màu xanh trên đầu node) nó sẽ kiểm soát việc một nhánh trong cây hoặc thậm chí một node có thể được thực thi hay không. Tôi đã sử dụng một công cụ decorator trong AI mà chúng ta sẽ thực hiện để yêu cầu nó cập nhật lên lộ trình khả dụng tiếp theo. Trong hình ảnh sau đây, bạn có thể lưu ý công cụ decorator **Attack & Destroy** xác định trạng thái trên đầu trang tổng hợp. Trạng thái này bao gồm hai tác vụ, **Attack Enemy** và **Move To Enemy**, cũng có một decorator yêu cầu nó chỉ thực thi khi trạng thái bot là **Search**:



Trong ảnh chụp màn hình trước, bạn có thể lưu ý công cụ decorator **Attack & Destroy** xác định trạng thái trên đầu trang tổng hợp. Trạng thái này bao gồm hai tác vụ, **Attack Enemy** và **Move To Enemy**, và **Move To Enemy** cũng có một công cụ decorator yêu cầu nó chỉ thực thi khi trạng thái bot là **Search**.

## Composite

Đây là những điểm khởi đầu của các trạng thái. Chúng xác định cách thức trạng thái sẽ hoạt động với luồng thực thi và trả giá trị. Chúng có ba loại chính: **Selector**, **Sequence** và **Simple Parallel**. Nhánh bắt đầu này có một câu lệnh điều kiện, nếu trạng thái bằng hoặc lớn hơn trạng thái **Search**:

**Selector** sẽ thực thi từng phần tử con của nó từ trái sang phải và không bị lỗi; tuy nhiên, nó trả về thành công khi một trong các con của nó trả về thành công. Vì vậy, nó tốt cho trạng



thái không kiểm tra các node được thực thi thành công. Ảnh chụp màn hình sau đây hiển thị một ví dụ về **Selector**:



Sequence sẽ thực thi các phần tử con của nó theo cách tương tự như Selector nhưng trả về thất bại khi một trong các phần tử con của nó trả về không thành công. Điều này có nghĩa là tất cả các node đều phải trả về thành công để hoàn thành chuỗi. Bạn có thể thấy một node Sequence trong ảnh chụp màn hình sau:

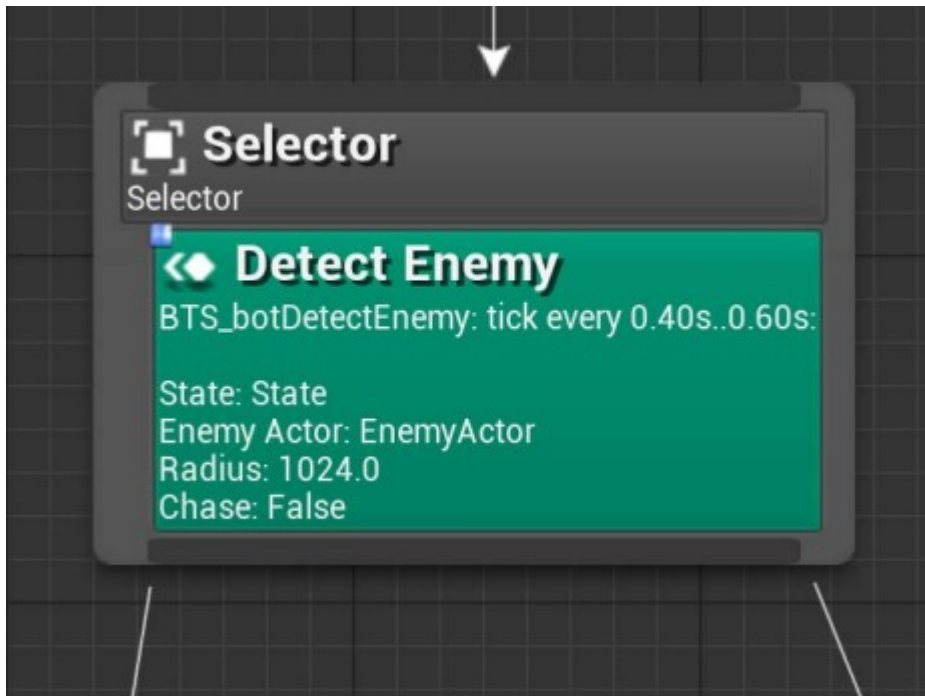


Cuối cùng nhưng không kém phần quan trọng, **Simple Parallel** cho phép bạn thực hiện một tác vụ và một cây về cơ bản cùng một lúc. Điều này rất tốt để tạo một trạng thái yêu cầu một tác vụ khác luôn được gọi. Để thiết lập nó, trước tiên bạn cần kết nối nó với một tác vụ mà nó sẽ thực thi. Tác vụ thứ hai hoặc trạng thái được kết nối tiếp tục được gọi với tác vụ đầu tiên cho đến khi tác vụ đầu tiên trả về thành công.

## Services

Các service sẽ chạy miễn là hỗn hợp mà nó được thêm vào vẫn được kích hoạt. Chúng nhấp tick vào các khoảng thời gian bạn đặt trong các thuộc tính. Chúng có một thuộc tính float khác gọi là **Tick Interval** cho phép bạn kiểm soát tần suất dịch vụ này được thực thi trong background. Các service được sử dụng để sửa đổi trạng thái của AI trong hầu hết các trường hợp vì nó luôn được gọi. Ví dụ: trong bot mà chúng ta sẽ tạo, chúng ta sẽ thêm một service vào nhánh đầu tiên của cây để nó được gọi mà không bị gián đoạn và có thể duy trì trạng

thái mà bot sẽ ở trong bất kỳ chuyển động nhất định nào. Nút màu xanh lục trong ảnh chụp màn hình sau đây là một dịch vụ có thông tin quan trọng một cách rõ ràng:



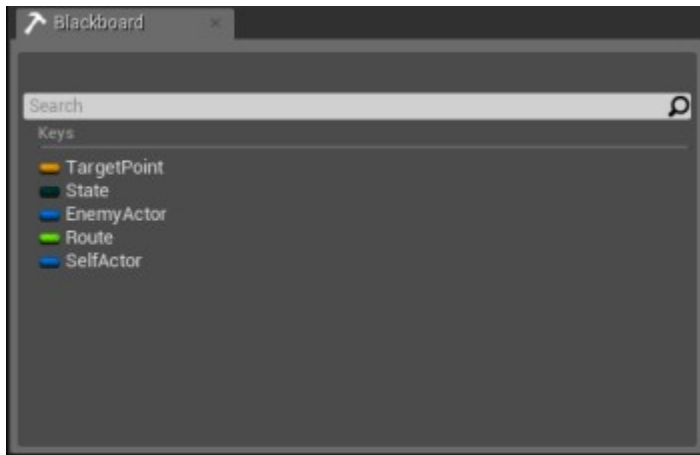
Service này, được gọi là **Detect Enemy**, thực sự chạy một chu kỳ lịch để cập nhật các biến Blackboard như **State** và **Enemy**.

## Tasks

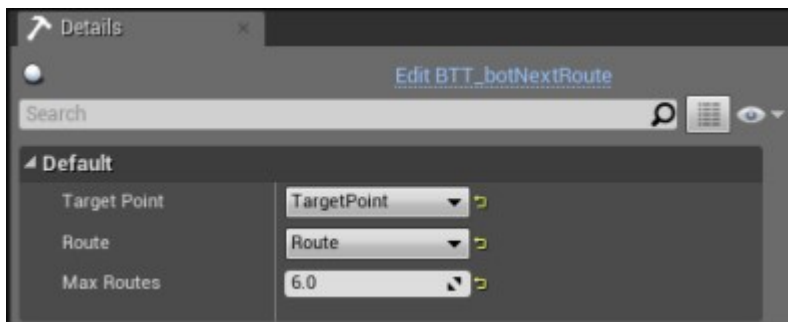
Tasks sẽ thực hiện công việc bắn thủ và báo cáo thành công hoặc thất bại nếu cần thiết. Chúng có các node kế hoạch chi tiết có thể được tham chiếu trong Behavior Tree. Có hai loại node mà bạn sẽ sử dụng thường xuyên nhất khi làm việc với Task: **Event Receive Excute**, nó sẽ nhận tín hiệu để thực thi các lệnh script được kết nối và **Finish Execute**, gửi tín hiệu trở lại và trả về đúng hoặc sai khi thành công. Điều này rất quan trọng khi thực hiện một tác vụ dành cho node tổng hợp **Sequence**.

## Blackboard

Blackboard là một tài nguyên lưu trữ các biến sẽ được sử dụng trong Behavior Tree của AI. Chúng được tạo bên ngoài Behavior Tree. Trong ví dụ của chúng tôi, chúng tôi sẽ lưu trữ một biến liệt kê cho trạng thái trong đối tượng **State**, **EnemyActor** để giữ kẻ địch hiện đang được nhắm mục tiêu và **Route** sẽ lưu trữ vị trí tuyến đường hiện tại mà AI được yêu cầu di chuyển tới, chỉ nêu tên một vài tuyến. Bạn có thể xem tất cả các biến hiện tại dưới dạng các khóa keys trong bảng Blackboard như sau:



Chúng hoạt động chỉ bằng cách đặt biến công khai của một node thành một trong các biến Blackboard có sẵn trong menu thả xuống. Quy ước đặt tên trong ảnh chụp màn hình sau giúp quy trình này được sắp xếp hợp lý:



## Hệ thống cảm quan

Một hệ thống cảm quan thường bao gồm một số mô-đun, chẳng hạn như thị giác (sight), âm thanh (sound) và bộ nhớ (memory), để giúp AI nắm bắt thông tin về môi trường. Một bot có thể duy trì cảm giác về trí thông minh bằng cách sử dụng âm thanh trong môi trường của chúng để thực hiện đánh giá rủi ro có chủ ý trước khi thực hiện một mối đe dọa nguy hiểm hoặc hỗ trợ một đồng đội gần đó đang kêu cứu. Việc sử dụng bộ nhớ sẽ cho phép bot tránh một khu vực mà nó nhớ đã nhìn thấy một mối đe dọa nghiêm trọng hoặc quay trở lại khu vực mà nó nhìn thấy nhóm của mình lần cuối. Việc tạo ra một hệ thống cảm giác trong trường hợp có người chơi kẻ thù chủ yếu dựa vào môi trường mà AI chiến đấu với người chơi. Nó cần có khả năng tìm chỗ ẩn nấp, trốn tránh kẻ thù, lấy đạn và các tính năng khác mà bạn cảm thấy để tạo ra AI nhập vai cho trò chơi của mình. Trò chơi với AI thách thức người chơi tạo ra trải nghiệm cá nhân độc đáo. Một hệ thống cảm giác tốt đóng góp thông tin quan trọng tạo nên AI phản ứng. Trong dự án này, chúng tôi sẽ sử dụng hệ thống cảm biến để phát hiện những con tốt mà AI có thể nhìn thấy. Chúng tôi cũng sẽ sử dụng các chức năng để kiểm tra đường ngắm của kẻ thù. Chúng tôi sẽ kiểm tra xem có một con tốt nào khác cản

đường chúng tôi hay không. Chúng tôi có thể kiểm tra vỏ bọc và các tài nguyên khác trong khu vực.

## Máy học (Machine Learning)

Học máy là một nhánh của riêng nó. Kỹ thuật này cho phép AI học hỏi từ các tình huống và mô phỏng. Đầu vào được lấy từ môi trường, bao gồm bối cảnh mà bot cho phép nó thực hiện các hành động quyết định. Trong học máy, các đầu vào được đặt trong một bộ phân loại có thể dự đoán một tập hợp các đầu ra với mức độ chắc chắn nhất định. Các bộ phân loại có thể được kết hợp thành các nhóm để tăng độ chính xác của dự đoán xác suất. Chúng ta sẽ không đào sâu vào chủ đề này, nhưng có rất nhiều tài nguyên để nghiên cứu về học máy, từ sách giáo khoa (Nhận dạng mẫu và Học máy của Christopher M. Bishop, Springer) đến các khóa học trực tuyến (Học máy trên coursera.org).

## Tracing (Dò tìm)

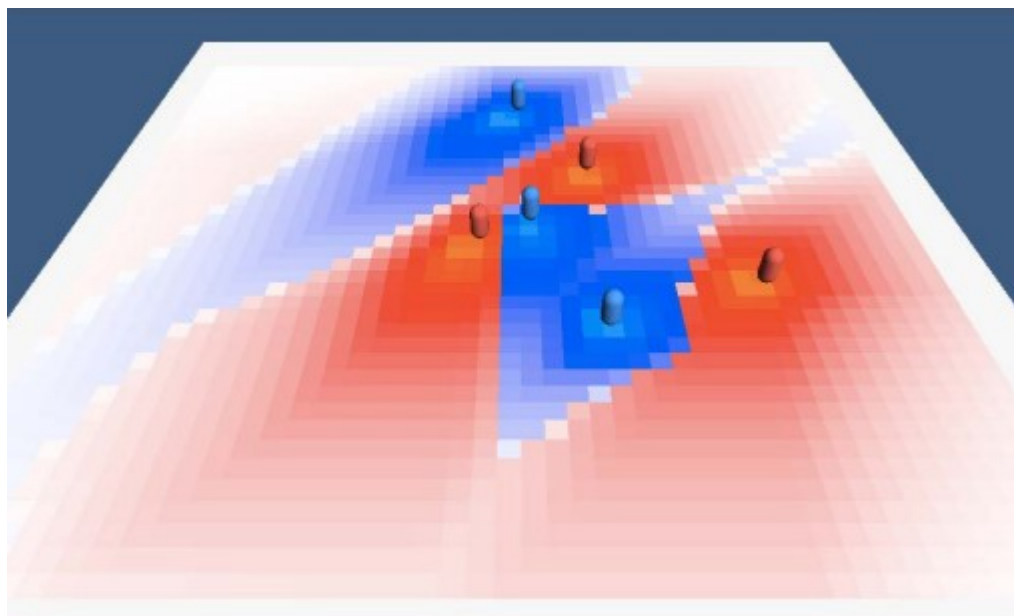
*Dò tìm* cho phép một tác nhân khác trong thế giới phát hiện các đối tượng bằng phương pháp ray tracing. Một đường thẳng dò tìm được gửi đi và nếu nó va chạm với một tác nhân, thì tác nhân đó sẽ được trả về cùng với thông tin về tác động. Dò tìm được sử dụng vì nhiều lý do; một cách nó được sử dụng trong FPS là phát hiện các hit testing. Bạn có quen thuộc với hitbox không? Khi người chơi của bạn bắn trong một trò chơi, một ray tracing sẽ được bắn ra và sẽ va chạm với hitbox của đối thủ, xác định thiệt hại cho người chơi và nếu bạn đủ khéo léo, nó sẽ dẫn đến cái chết. Các hình dạng khác có sẵn cho ray tracing, chẳng hạn như hình cầu, viên nang và hộp, cho phép tạo ray tracing cho các tình huống khác nhau. Gần đây, tôi đã sử dụng Box Trace cho ô tô của mình để phát hiện các vật thể ở gần nó.

## Bố trí tầm ảnh hưởng

Influence Mapping (Bố trí tầm ảnh hưởng) không phải là một cách tiếp cận hữu hạn; ý tưởng là các vị trí cụ thể trên bản đồ sẽ được quy về thông tin ảnh hưởng trực tiếp đến người chơi hoặc AI. Một ví dụ về việc sử dụng Influence Mapping với AI là presence falloff. Giả sử chúng ta có AI kẻ thù khác trong một nhóm; presence map của chúng sẽ tạo ra một vòng tròn hướng tâm xung quanh nhóm với cường độ dựa trên quy mô của nhóm. Bằng cách này, AI khác khi đi vào khu vực này sẽ biết rằng chúng đang đi vào khu vực do AI của kẻ thù khác chiếm giữ.

Thông tin thực tế không phải là mục đích duy nhất mà mọi người sử dụng, vì vậy chỉ cần hiểu rằng thông tin đó nhằm mục đích cung cấp một cấp độ đầu vào khác để giúp bot của bạn đưa ra nhiều quyết định bổ sung hơn. Như được hiển thị trong hình ảnh sau đây, các

màu khác nhau biểu thị các vùng được chiếm giữ bởi các loại AI khác nhau và cường độ màu biểu thị mức độ ảnh hưởng đối với từng nhân vật AI:



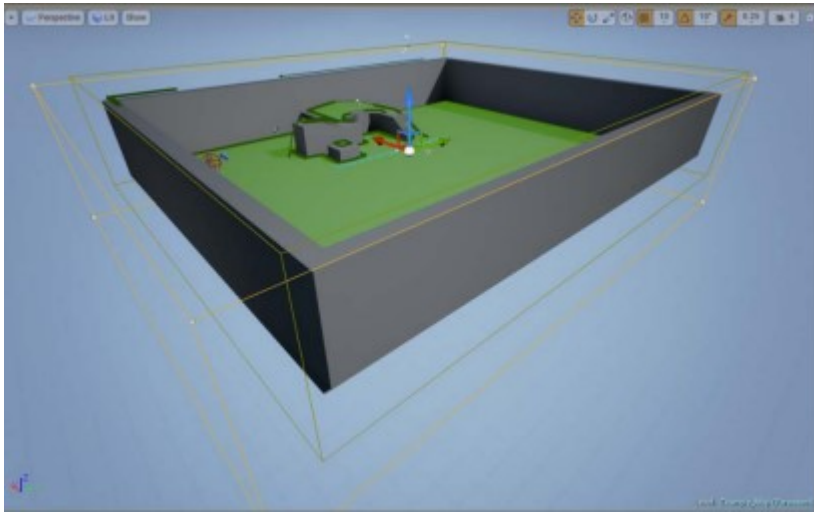
*Thông tin thực tế không phải là mục đích duy nhất mà mọi người sử dụng, vì vậy chỉ cần hiểu rằng thông tin đó nhằm mục đích cung cấp một cấp độ đầu vào khác để giúp bot của bạn đưa ra nhiều quyết định bổ sung hơn.*

## Các công cụ trong Unreal Engine 4

Unreal Engine 4 cung cấp một bộ công cụ hoàn chỉnh để thêm tính năng AI phổ biến vào trò chơi của bạn. Chúng ta sẽ đi vào chi tiết của từng công cụ trong tài liệu này. Dưới đây là danh sách các công cụ được đưa vào:

- **Behavior Tree:** Nó được sử dụng để tạo các trạng thái khác nhau và logic đằng sau AI.
- **Navigation Component:** Phần này xử lý chuyển động cho AI.
- **Blackboard Asset:** Chúng được sử dụng để lưu trữ thông tin. Chúng đóng vai trò là biến cục bộ cho AI.
- **Enumeration:** Điều này được sử dụng để tạo các trạng thái mà bạn có thể thay thế giữa chúng.
- **Target Point:** Waypoint class của chúng ta bắt nguồn từ Target Point class mà chúng ta sẽ sử dụng để tạo một dạng cơ bản của Path node.
- **AI Controller và Character:** Controller này sẽ xử lý giao tiếp giữa thế giới và con tốt được điều khiển cho AI.
- **Navigation Volume:** Điều này được sử dụng để tạo Navigation Mesh trong môi trường để bật tính năng Path Finding cho AI.

Hãy xem màn hình sau:



Có hai loại volume NavMesh. Đầu tiên, khối lượng NavMesh Bounds, xác định khu vực cho NavMesh. Volume của Nav Modifier, khi được cung cấp cùng với một Nav Area class, sẽ ảnh hưởng đến các thuộc tính điều hướng của volume NavMesh Bound nơi hai phần giao nhau.

## Tóm tắt

Trong chương này, chúng ta bắt đầu bằng cách giới thiệu AI trong trò chơi và thảo luận về lý do tại sao AI lại quan trọng đối với trải nghiệm người dùng khi chơi trò chơi của chúng ta. Sau đó, chúng ta minh họa hầu hết các kỹ thuật AI trong trò chơi đã sử dụng và khả năng của chúng. Các công cụ UE4 tương ứng dành cho AI trong trò chơi cũng được đề cập để cung cấp một bức tranh toàn cảnh hơn về nội dung mà chúng tôi sẽ trình bày xuyên suốt tài liệu này. Trong chương tiếp theo, chúng ta sẽ tạo AI cơ bản bằng cách thiết lập một player do AI điều khiển và thêm một số hành vi đơn giản vào đó.