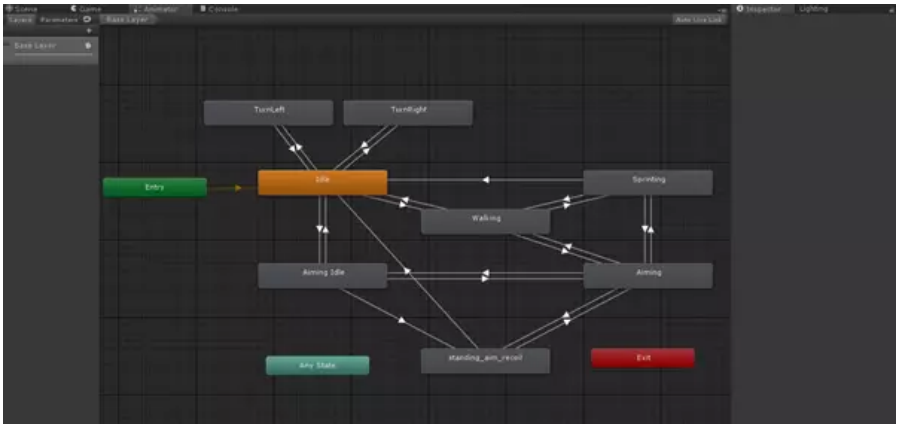
Unity - Cơ bản Mecanim

Một trong những yếu tố quan trọng vì sao game hấp dẫn người chơi là cử động, tương tác và thao tác các của nhân vật đối với thế giới và các nhân vật khác, những cử động đó là hoạt hình, và Unity với hệ thống điều khiển hoạt hình Mecanim giúp cho việc điều khiển chúng một cách dễ dàng hơn nhiều.



Mecanim là gì?

Mecanim là hệ thống State Machine của Unity, nói đơn giản là nó cho phép người dùng tạo những "trạng thái" (state) để chạy những hoạt hình khác nhau và định nghĩa các logic để chuyển đổi giữa các trạng thái khác nhau.

Có thể làm gì với Mecanim?

Mecanim có thể sử dụng khi import các file .fbx với rig của chúng, hoặc các hoạt hình 2D được tạo trong Unity. Mesh và Rig có thể là bất kỳ loại (người, động vật hay tĩnh vật). Rig cần phải được định nghĩa là Humanoid (loài người) hoặc generic (các loại khác) vì Mecanim không hoạt động được với những hoạt hình thuộc loại legacy.

Sự khác biệt giữa hoạt hình Humanoid và Generic?

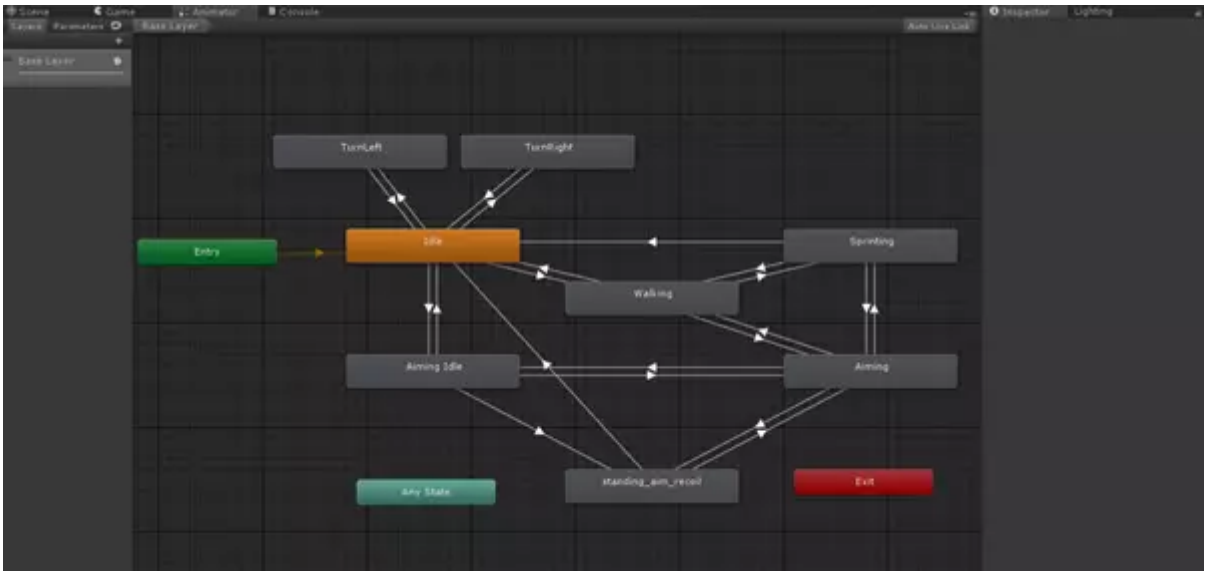
Nếu import hoạt hình với object có 2 tay, 2 chân và 1 đầu thì thông thường object đó được cho là Humanoid rig. Các hoạt hình liên quan đến humanoid có thẻ hoạt động với bất kỳ loại rig nào cùng loại, là Humanoid, kể cả khi có sự khác biệt về rig. Với bất kỳ loại rig nào không phải Humanoid, hoạt hình cần phải import dưới dạng generic, hoạt hình generic sẽ chỉ dùng được với các generic rig.

Các bộ phận của Mecanim

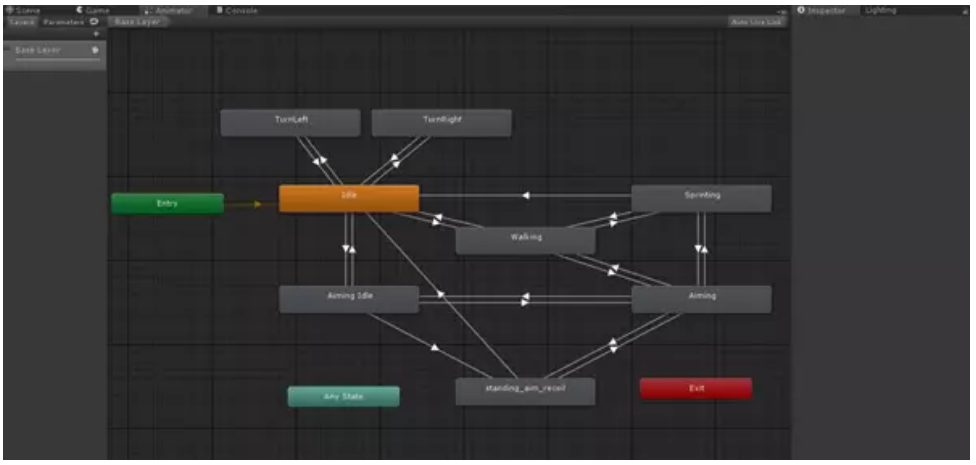
Animator

Animator là component đặc biệt được gắn với các nhân vật có hoạt hình. Nó được tự động gắn với tất cả prefab tạo từ file .fbx được import dưới dạng Humanoid hoặc Generic. Vai trò Animator là chuyển thông tin từ game object vào controller hoạt hình, nó có một số property

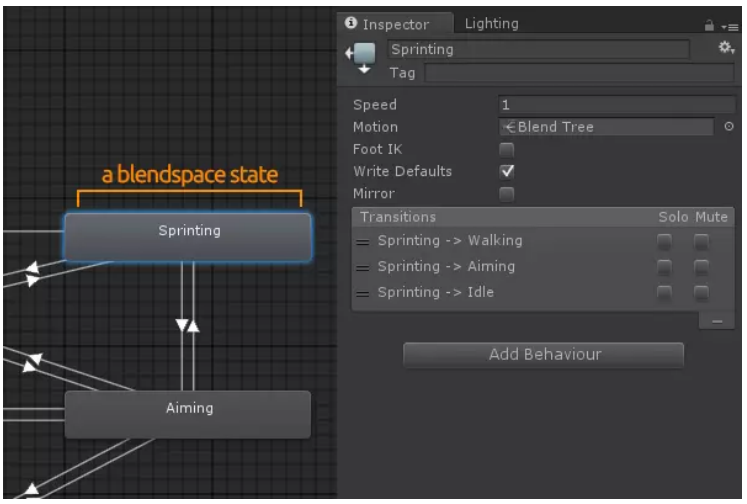
sau đây:



Animator Controller - Controller hoạt hình dưới dạng node-base để định nghĩa các trạng thái và vận chuyển từ trạng thái này sang trạng thái khác.
Avatar - Về cơ bản là định nghĩa của rig. Có thể dùng để thay đổi hoạt hình với rig tương đương (Humanoid hoặc Generic). Khi áp dụng .fbx vào object componenet Avatar sẽ được tự động gắn vào nó.
Apply Root Motion - Với chức năng này Mecanim sẽ dùng root motion (cử động chính) của nhân vật để di chuyển xung quanh scene. Tuy nhiên nếu có sự cần thiết điều chỉnh cử động nhân vật chỉ bằng code thì nên tắt chức năng này, tuy nhiên root motion mang lại cử động tự nhiên và mượt mà hơn.



Animator Controller
Animator controller là component tất yếu của Animator, animator controller có thể xem được khi chọn object mà nó được gắn lên thông qua cửa sổ Window - Animator. Animator Controller là nơi mà các trạng thái và logic của các hoạt hình được tạo ra.



State

Trạng thái trong mecanim là một nút đơn và có thể có vài dạng. Chọn vào trạng thái sẽ hiển thị những thông tin có ích như hoạt hình và blend tree được dùng bởi state đó, với tình trạng Mirrored của hoạt hình, danh sách chuyển đổi trạng thái với những trạng thái khác. Sau đây là 1 số trạng thái có thể được tạo:

Animations - Trạng thái này có thể là một hoạt hình đơn, để tạo ra Trạng Thái Animation chỉ cần kéo hoạt hình của mình vào cửa sổ Animator. Trạng thái này sẽ có những property như di chuyển (hoạt hình được dùng), tốc độ (tốc độ hoạt hình được chạy) và Mirror (quay ngược cử động).

Blendspaces - Dùng để điều khiển nhiều animation khác nhau trong một nút, với hai thể loại blendspace là một chiều hoặc hai chiều. Ví dụ kinh điển dùng blendspaces là vận động cơ bản, blendspace có thể dùng để điều khiển tốc độ đi thẳng, sang trái , phải, quay ngược 45 độ hoạt hình trong một nút.

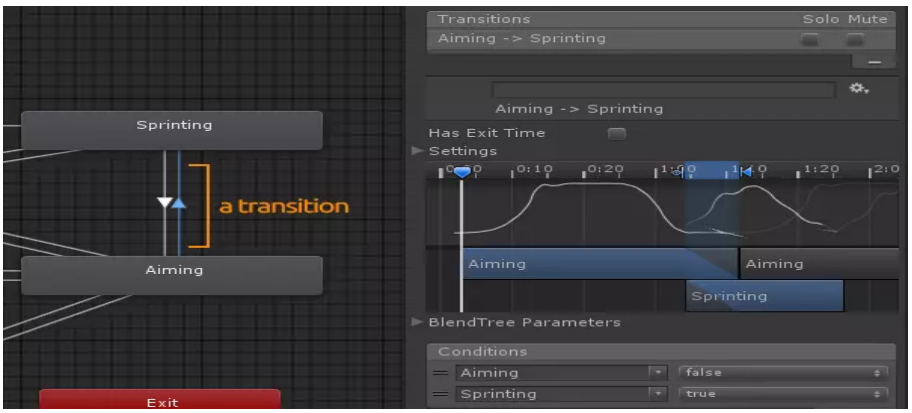
Sub-Statemachines - Có thể dùng với các trạng thái khác biệt nhau nhưng có chung logic. Ví dụ tốt nhất là di chuyển cơ bản và hành động trong trạng thái thường và trạng thái ẩn nấp. Trạng thái ẩn nấp có thể được đưa vào sub-state machine và nó sẽ có logic riêng, hành động và điều khiển.

Default State - Mỗi animator cần có default state, trạng thái ban đầu, nút có màu da cam đặc biệt, đây sẽ là trạng thái đầu tiên khi khởi động state machine.

Any State - Một state rất hữu dụng, cho phép di chuyển đến trạng thái này từ bất kỳ trạng thái khác, rất phù hợp để chạy hoạt hình mà sẽ cắt đứt tất cả trạng thái khác. Như khi một nhân vật hoạt động bình thường nhưng bị sát thương, sẽ bắt buộc rơi vào trạng thái bị thương hoặc chết.

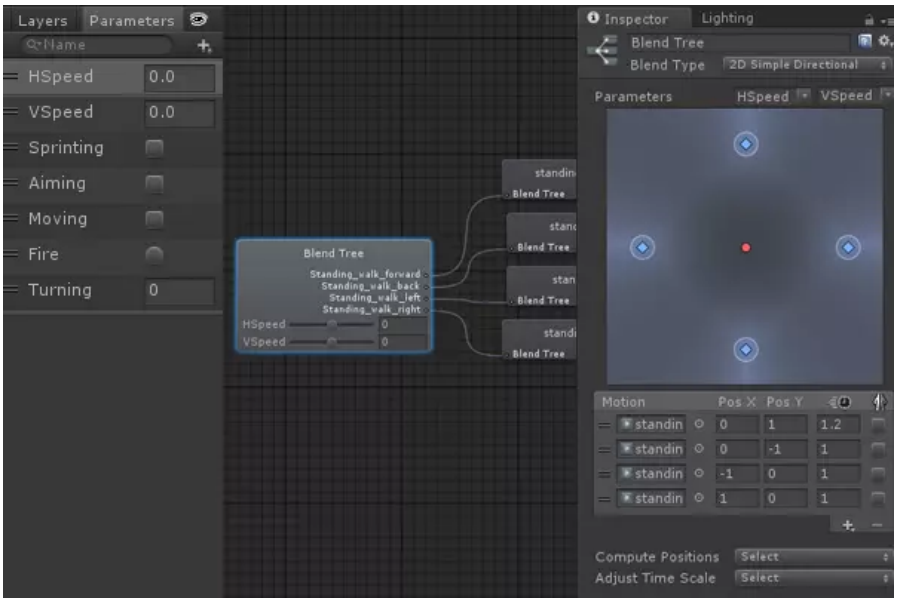
Entry (Unity 5) - Với Unity 5 nếu dùng sub-statemachine các trạng thái Entry sẽ thông báo khi nào trạng thái đó được di chuyển đến từ các trạng thái khác.

Exit (Unity 5) - Tương tự như Entry, nhưng Exit sẽ thông báo khi trạng thái này chuyển sang trạng thái khác.



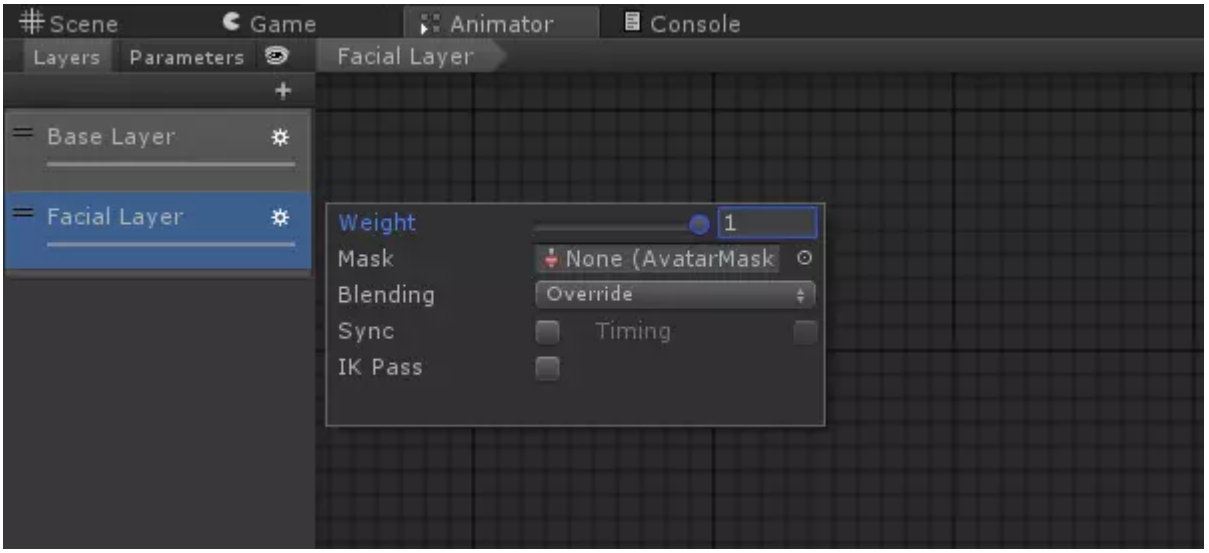
Transitions (chuyển trạng thái)
Để có thể chuyển được từ trạng thái này sang trạng thái kia cần tạo transition. Transition được tạo ra bởi click chuột phải vào nút bất kỳ và chọn "Create Transition". Một mũi tên sẽ được tạo ra, hãy kéo mũi tên đó đến nút cần chuyển để tạo ra "sự kết nối", để có thể có được sự di chuyển từ nút này sang nút kia cần tạo các điều kiện khá nhau, trong inspector. Sau đây là những điều kiện:

Exit Time - Dùng để xác định thời điểm chạy (%) hoạt hình để thoát khỏi trạng thái tự động khi hoạt hình chạy xong, ví dụ, nếu chỉnh số % là 90, thì sau khi hoạt hình chạy được 90% nó sẽ tự động chuyển sang trạng thái tiếp.
Parameters - Các parameter như boolean, floats hay int, với các phép tính như lớn hơn, nhỏ hơn hay bằng, khi các điều kiện được thỏa mãn trạng thái của animation sẽ được thay đổi.
Has Exit Time (Unity 5) - Trong Unity 5, Mecanim đã được cải thiện đáng kể, thay bằng dùng Exit Time Unity 5 có parameter mới được gọi là Has Exit Time, nếu Has Exit Time là True và không có các điều kiện để thay đổi trạng thái khác, thì animation sẽ tự đổng thay đổi sau exit time. Trong trường hợp Has Exit Time là true và có các điều kiện cần được gặp, mecanim sẽ chờ đến khi các điều kiện đó được đáp ứng và sau khi hoạt hình chạy xong, mới chuyển sang trạng thái tiếp theo. Tuy nhiên nếu có nhưu cầu chuyển hoạt hình ngay lập tức khi điều kiện đủ mà không cần chờ đến khi hoạt hình chạy hết, hãy chuyển Has Exit Time thành false.



Parameters
Để tạo ra parameter, hãy mở animation controller, bên phải có thể nhìn thấy parameter tab, tại đây hãy ấn nút "+" và parameter mới sẽ được tạo, dưới đây có thể thấy những parameter như HSpeed và VSpeed với giá trị float (0.0). Những parameter này có thể được dùng để điều khiển blendtree. Sau đây là những parameter có thể dùng:

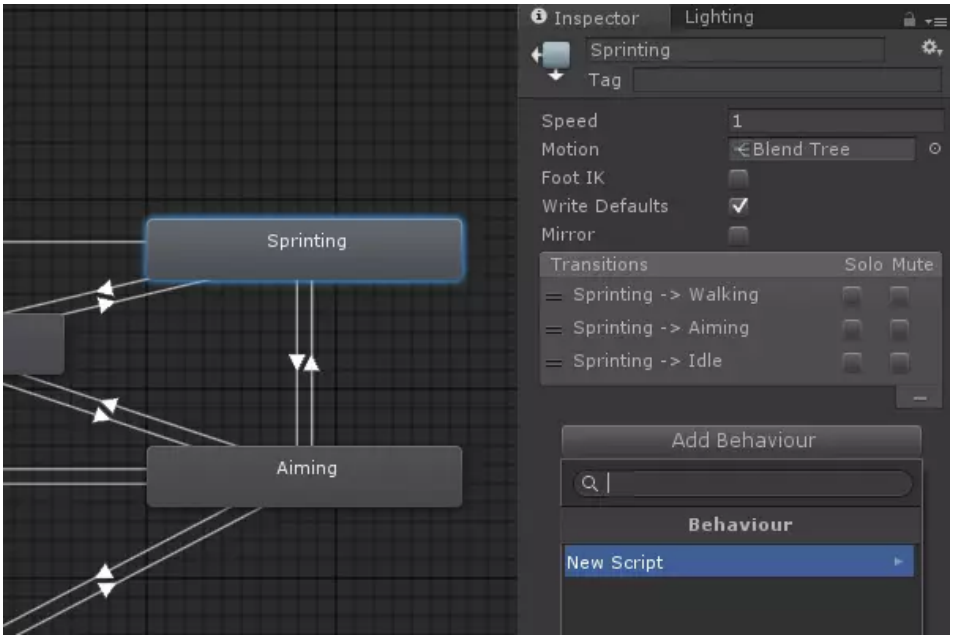
Float - Với giá trị là số thập phân, có thể dùng như để điều khiển tốc độ di chuyển giữa hai trạng thái là chạy hay đi bộ.
Integer - Số nguyên, có thể dùng để xác định liệu trạng thái có thể được di chuyển hay không tùy vào giá trị.
Bool - Với giá trị true, false, dùng để tạo ra những điều kiện cho phép chuyển đổi trạng thái, như "chạy", khi "chạy" là true, trạng thái sẽ được thay đổi, bool là parameter rất phù hợp với những hoạt hình cần loop.
Trigger - Trigger là bool với trạng thái luôn luôn là true, nhưng không như bool nó chỉ được dùng đúng một lần, trong khi bool sẽ loop hoạt hình liên tiếp chừng nào nó không được chuyển thành giá trị khác. Trigger sẽ chỉ dùng một lần đề chuyển trạng thái và ngừng lại.



Layers
Mecanim sở hữu hệ thống tạo lớp cho các hoạt hình, chúng có thể dùng để đè lên các hoạt hình khác trong tại một hoặc nhiều bộ phận trên cơ thể. Cách mô tả dễ hiểu nhất là như cần điều khiển các bộ phận cơ thể riêng, như nhân vật có thể vừa di chuyển vừa dùng những cử chỉ tay, chân khác nhau cùng lúc. Layer có vài thuộc tính như sau:

Mask - Có khả năng điều khiển các bộ phận đơn trên cơ thể object, tùy vào mình chọn bộ phận nào sẽ được mask. Như nếu chọn tay, object có thể vừa cử động vừa dùng những hoạt hình tấn công với tay.
Blending - Để đặt layer thành additive(hoạt hình sẽ được đưa vào hoạt hình cơ bản) hoặc override (animation sẽ được thay thế cho hoạt hình cơ bản). Override thông thường sẽ dùng nhiều hơn, additive cần có sự hiểu biết cao hơn để dùng nó một cách hiệu quả.
Sync (Unity Pro) - Dùng để tạo ra nhiều layer giống nhau nhưng có cách hoạt động chút khác nhau. Như nhân vật có thể có hoạt hình bay và chạy, mặc dù là hai hoạt động khác

nhau nhưng cùng logic điều khiển, có thể dùng Sync để khớp hai layer animation với nhau mà vẫn dùng code điều khiển logic.
Weight - Xác định độ nặng qua code (Animator.SetLayerWeight(String, Float)), đơn giản nó sẽ xác định layer sẽ có tác động thế nào với các layer còn lại, layer với weight lớn hơn sẽ đè lên layer có weight thấp hơn.



Animator Scripting
Các giá trị trong animator controller có thể lấy được trong code, để điều khiển chúng một cách linh hoạt hơn, Unity đã tạo những function trong API sau đây để giúp làm việc với Mecanim dễ hơn.

GetComponent<Animator>() - Dùng để lấy ra componenet Animator của Object.
Animator.SetInteger(String, Int) - Để áp dụng lên trạng thái(tên string) với giá trị (số int)
Animator.SetFloat(String, Float) - Tương tự như SetInt() nhưng với float.
Animator.SetBool(String, Bool) - Truyền giá trị bool lên trạng thái (tên string).
Animator.SetTrigger(String) - Truyền giá trị mặc định lên trạng thái (tên string).
Trong Unity 5 đội ngũ Unity đã mở rộng API của Mecanim, cho phép làm việc với Mecanim đơn giản và hiệu quả hơn, có thể đưa script trực tiếp vào Mecanim, giờ đây Mecanim không chỉ sử dụng với hoạt hình mà có thể áp dụng hoàn toàn cho logic code, cho phép chuyển đổi trạng thái của game object này sang khác, thành bước phá lớn cho Unity. Sau đây là những function mới trong API:

OnStateEnter() - Đây là function sẽ được tự động gọi khi trạng thái được khởi động, tại khung hình đầu tiên.
OnStateUpdate() - Được gọi với mỗi khung hình trong lúc trạng thái được chạy.
OnStateExit() - Được gọi tại khung hình cuối cùng trước khi thoát khỏi trạng thái.
OnStateIK() - Để điều khiển IK tại thời điểm trạng thái đang chạy.
Nhờ Unity 5 Mecanim đã trở thành state machine tuyệt vời dùng để điều khiển logic game, không chỉ riêng với hoạt hình, khiến Unity gần hơn với các engine game lớn khác, giảm thiểu thời gian xử lý logic trong code rất nhiều. Với người sử dụng Unity 5, Mecanim trở nên kiến thức tối thiểu cần biết.