Làm Game 2D bằng Unity - Phần 3 - Animation và điều khiển hành động nhân vật (Animator)

20 January 2014 at 22:43

Chào các bạn,

Ở phần trước chúng ta đã biết cách tạo các game object, và các sprite cho các game object. Ở phần này chúng ta sẽ tìm hiểu cách tạo một animation và điều khiển các hành động của animation đó ví dụ như đi, đứng, chạy, nhảy...

Trước khi đi tiếp, nếu bạn nào chưa đọc các phần trước, ta có thể tham khảo lại ở đây

Link bài viết trước: https://www.facebook.com/notes/hội-lập-trình-viên-game-đà-nẵng/làm-game-2d-bằng-unity-phần-2-tạo-dự-án-game-2d-sprite-và-gameobject/232425046929765

Dự án mẫu cho phần 3 ở đây: https://www.dropbox.com/sh/b9uear8vzocr6kj/-SEOHSwc_c

III. Tạo các đối tượng cơ bản

1. Game Object

2. Sprite

3. Animation

Một animation là một hình ảnh động mô tả một đối tượng nào đó trong game.
Ví dụ: có thể là một chiếc xe đang chạy, hay một nhân vật đang đi....
Một animation trong Unity có thể bao gồm nhiều hành động, một hành động như vậy gọi là một clip.
Ví dụ: một nhân vật có thể có các hành động đi, đứng, nhảy....
Có hai kỹ thuật để tạo animation (cả 2D và 3D): đó là kỹ thuật key frame và kỹ thuật skeletal hay spine.

a. Kỹ thuật tạo animation

*** Kỹ thuật key frame**

Đối với kỹ thuật key frame, người ta sử dụng một sprite cho một key frame của hành động.

Mỗi sprite là một keyframe

Để tạo ra chuyển động, ta sẽ vẽ một key frame tại thời điểm đầu và thay đổi tuần tự các key frame sau, chúng ta sẽ có được một animation.

Đây là phương pháp đơn giản nhất để tạo chuyển động, nhưng lại tốn kém về bộ nhớ, vì ta phải tốn nhiều sprite cho nhiều chuyển động khác nhau.

*** Kỹ thuật skeletal hay spine hay bộ xương**
Đối với kỹ thuật này, người ta chia đối tượng ra thành nhiều sprite, mỗi sprite là một bộ phận của đối tượng (giống như 1 khúc xương của bộ xương). Để tạo ra một key frame mới, ta sẽ thay đổi các sprite về vị trí, độ lớn, xoay của các sprite thành phần có liên quan đến chuyển động. Sau đó kết hợp các key frame lại với nhau như kỹ thuật key frame để tạo thành các animation.

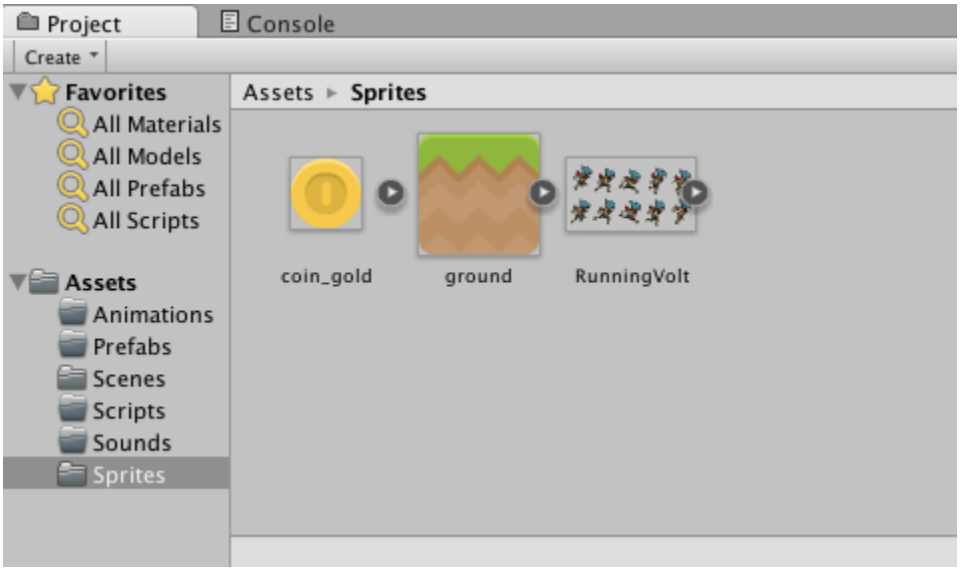
Chúng ta có thể xem các sprite cấu tạo nên một key frame ở ảnh dưới.

2D Spine

Cách này có vẻ tốn thời gian hơn, nhưng lại rất là hiệu quả, đặc biệt là tiết kiệm được nhiều bộ nhớ. (*Ngoài lề một tí thì nếu trong một quy trình làm game chuyên nghiệp, cách tạo các animation này thuộc về vai trò của game designer, chứ không phải của lập trình viên. Có thể bạn là một trình viên nếu bạn đang đọc bài viết này :D*)

b. Tạo animation bằng Unity

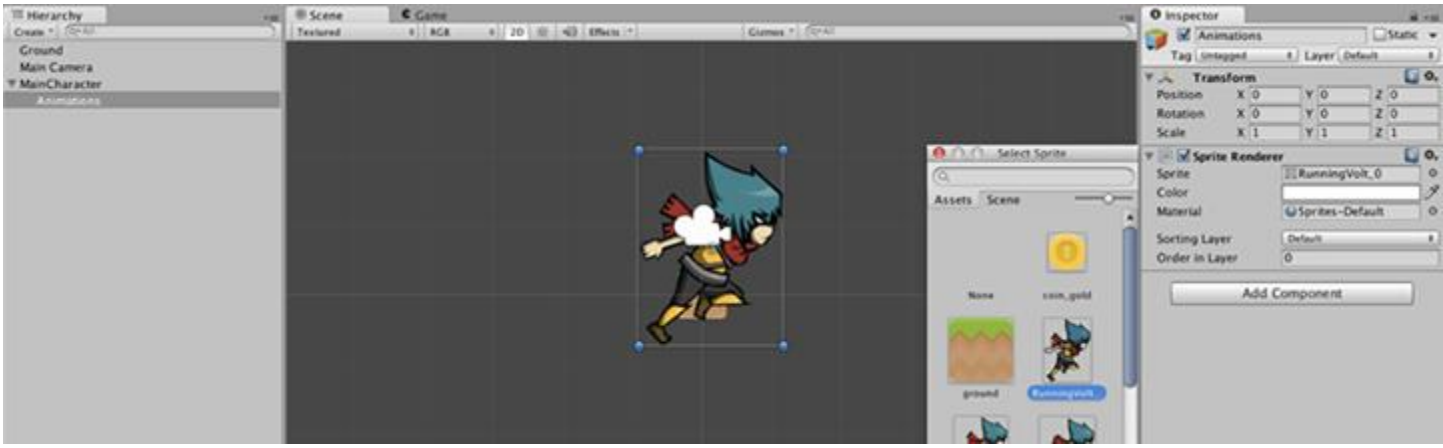
Do sự giới hạn về tài nguyên và thời gian nên trong bài viết này mình chỉ hướng dẫn cách tạo animation theo kỹ thuật Key Frame. Cách tạo animation theo skeletal cũng tương tự.
Từ phần trước chúng ta đã tạo được những Sprite cơ bản như sau:



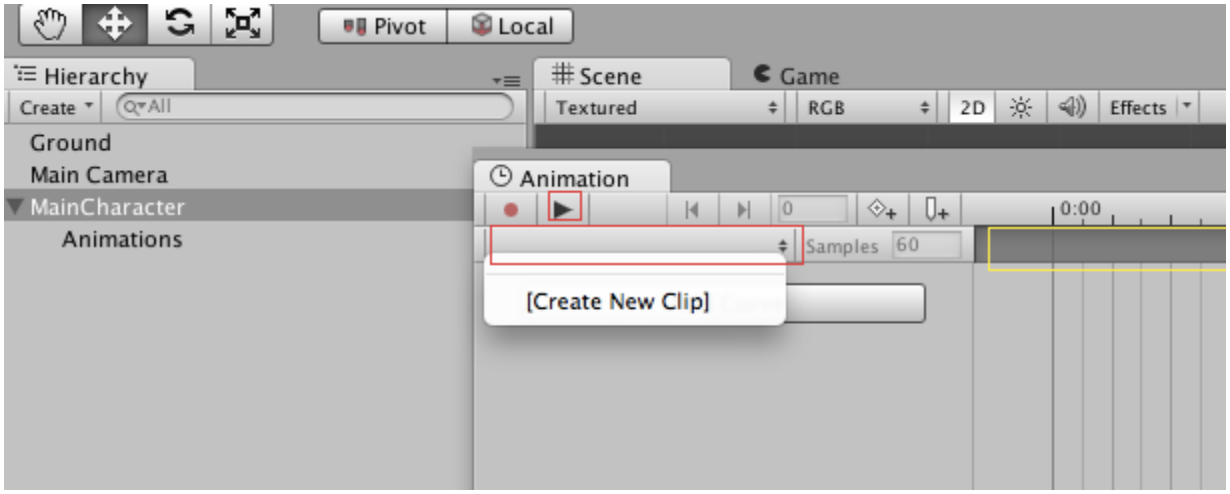
Bước 1: Tạo một Empty GameObject đặt tên là MainCharacter (Parent Object)

Bước 2: Tạo một đối tượng Empty GameObject nữa, đặt tên là Animations, là đối tượng con của MainCharacter. (Đối tượng con nên đặt ở vị trí 0,0,0).

Bước 3: Thêm Sprite Render cho đối tượng con Animations vừa tạo. Rồi chọn sprite hiển thị mặc định cho Animation này. Kết quả như sau:



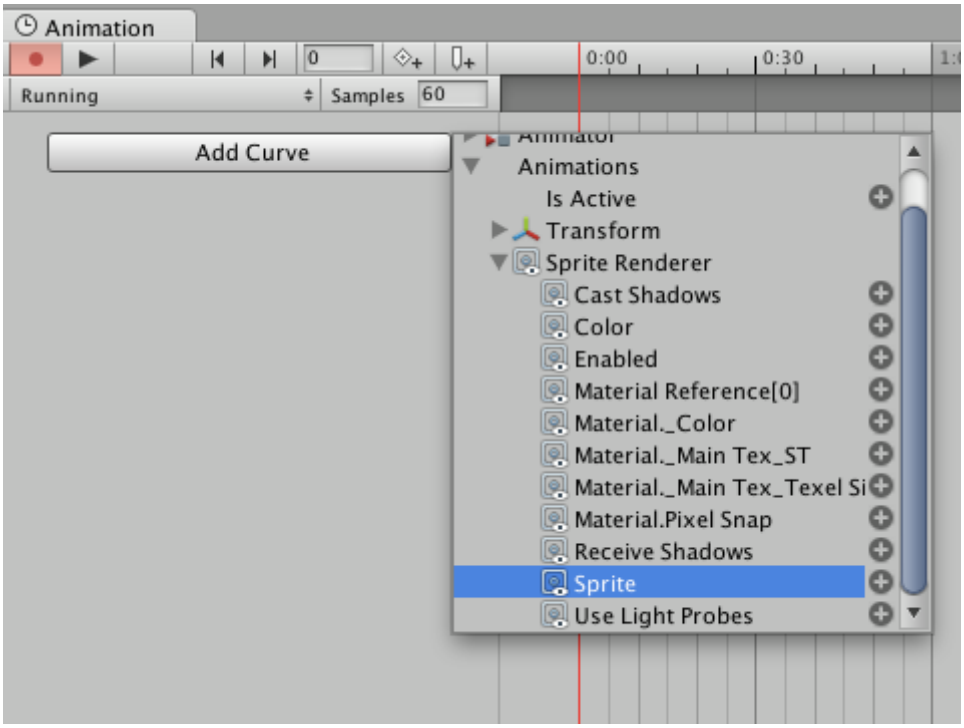
Bước 4: Chọn đối tượng MainCharacter ở cửa sổ Hierarchy, rồi chọn Menu -> Window -> Animation
Một cửa sổ Animation editor hiện ra như sau:



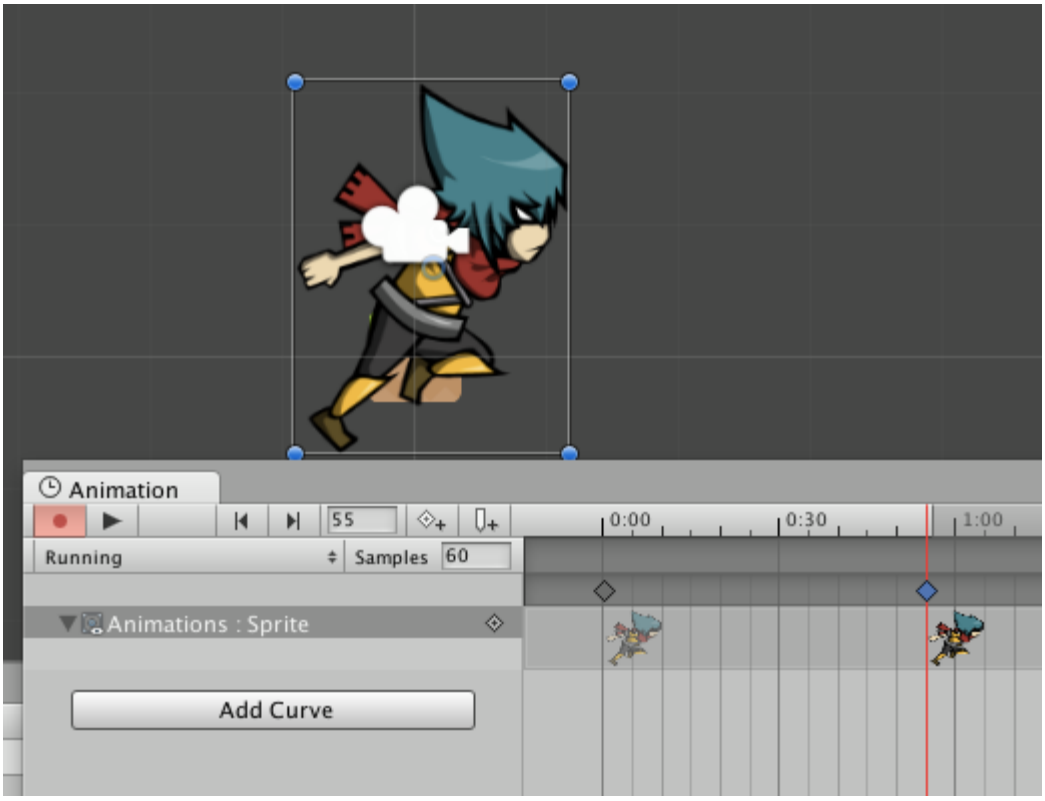
* Hình chữ nhật đỏ nhỏ: nút play để xem trước animation

* Hình chữ nhật đỏ lớn: danh sách các clip hiện thời của aniamtion
* Hình chữ nhật vàng: thanh key frame
Đầu tiên ta sẽ click vào danh sách clip rồi chọn Create New Clip, ta đặt tên clip là Running, rồi save lại ở thư mục Animations của Assets.

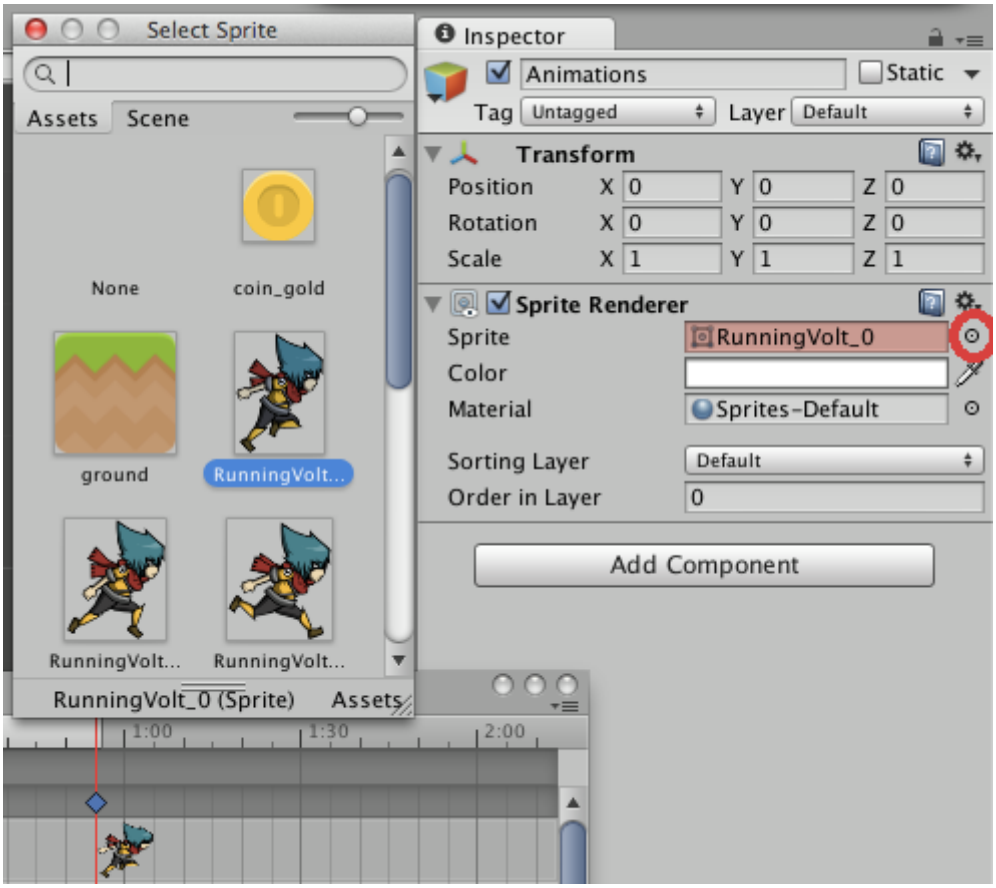
Bước 5: Ở cửa sổ Animation Editor, chọn Add Curve, chọn Animations (**Đối tượng con của đối tượng MainCharacter**) chọn Sprite Render, chọn Sprite. (Đối với một clip bất kỳ, bước này bắt buộc phải có).



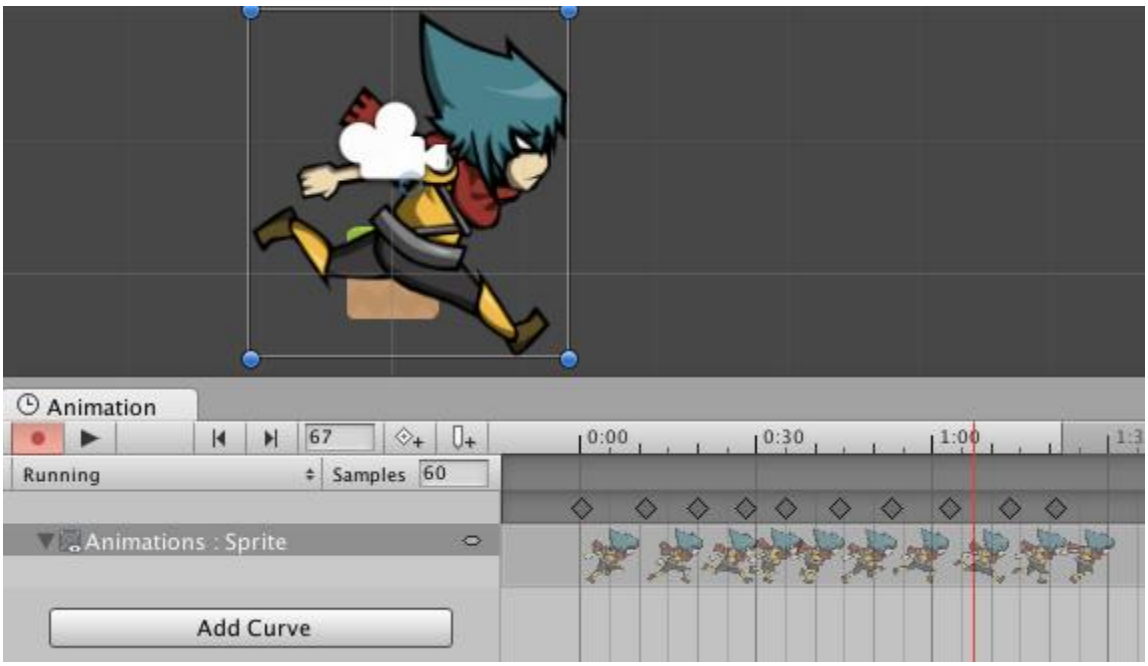
Kết quả như sau, mặc định sẽ tạo ra tối thiểu là 2 Key Frame.



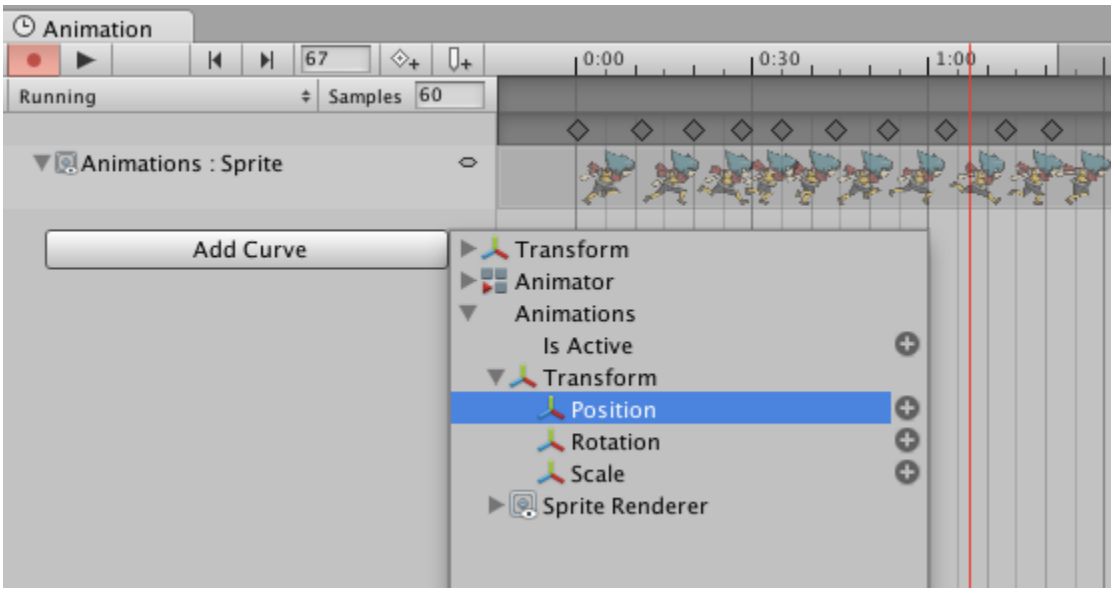
Tiếp theo ta chọn key frame thứ 2, ở cửa sổ **Inspector**, ở component **Sprite Render**, ta tiến hành đổi sprite khác (RunningVolt1 thay vì RunningVolt0). Chọn nút được bao quanh bởi ô tròn đỏ, rồi chọn Sprite khác từ cửa sổ mới hiện ra.



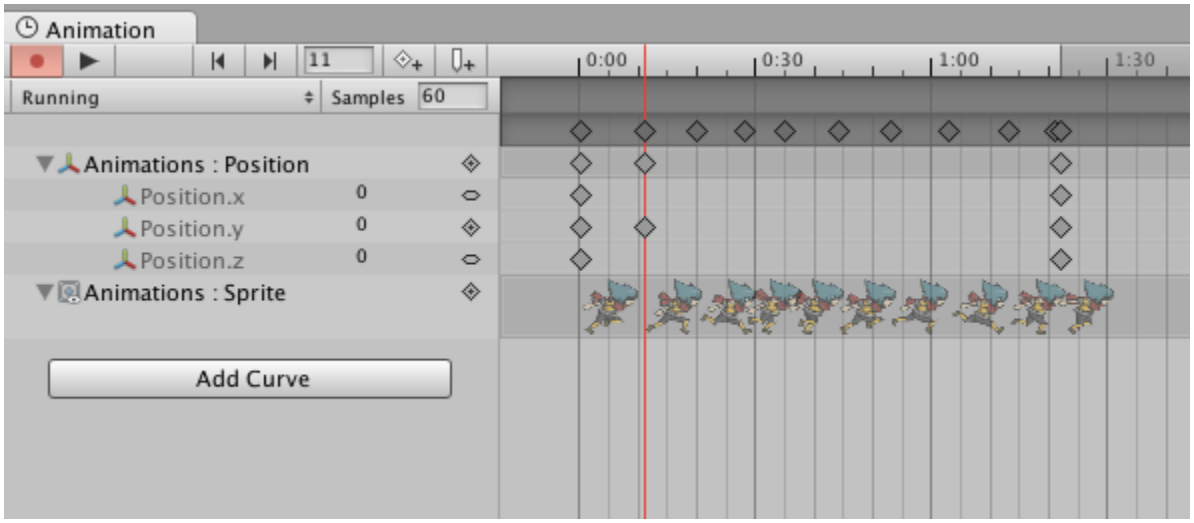
Bây giờ, chúng ta chỉ cần click đúp vào thanh Key Frame (Thanh có ô vuông màu vàng ở hình trước) để thêm các key frame và kéo thả các key frame sao cho thời gian phù hợp để có được chuyển động cần thiết.



Ngoài thay đổi sprite chúng ta có thể thay đổi Transform (Translate, Scale, Rotation) cho sprite tại mỗi key frame, bằng cách thêm Curve Transform cho đối tượng Animation như hình:



Sau đó ta chỉ việc chọn các key frame, rồi đặt các giá trị transform cho phù hợp theo ba trục x, y, z.



Tương tự ta sẽ tạo các clip Jump (nhảy), Idle (trạng thái nghỉ) cho đối tượng.

Chú ý:
- Ta sẽ thay đổi transform và sprite render của đối tượng con Animation chứ không thay đổi transorm và sprite render của đối tượng cha là MainCharacter.
Giải thích: animation sẽ làm thay đổi transform của sprite, nên nếu thêm trực tiếp vào đối tượng cha, sau này thêm thành phần vật lý, hoặc áp dụng các phép transform vào sẽ bị sai, hoặc mất tác dụng.

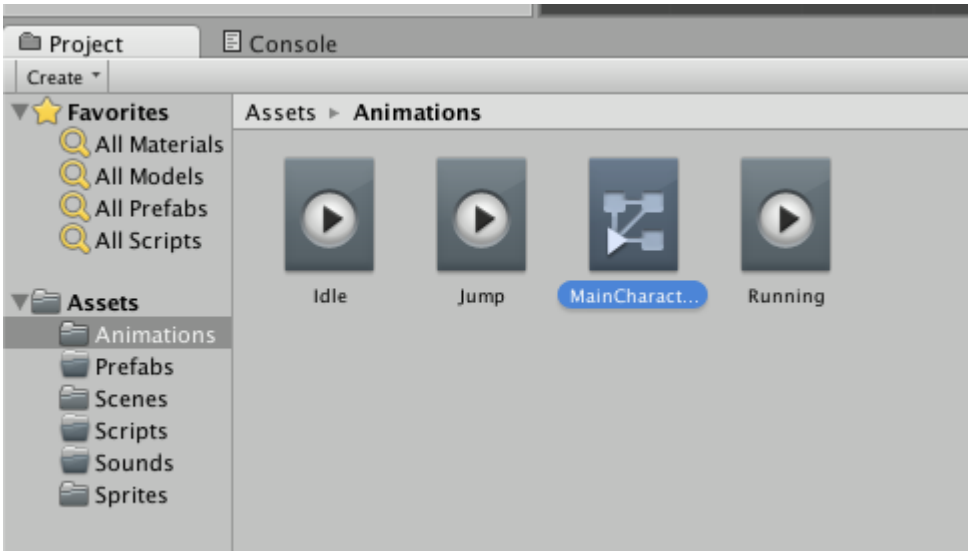
- Đối với các hành động nhảy chúng ta sẽ không thay đổi vị trí các key frame, vì làm như vậy khi thêm thành phần vật lý vào đối tượng cha, lúc nhân vật nhảy, hình ảnh của nó nhảy lên nhưng vị trí của nó để tính vật lý (theo đối tượng cha) vẫn nằm ở dưới hoặc thấp hơn hình ảnh --> Không đúng thực tế.
Đối với hành động nhảy này, ta chỉ cần chọn sprite đang ở tư thế nhảy mà thôi.

- Một Clip bất kỳ, phải có tối thiểu 2 key frame.

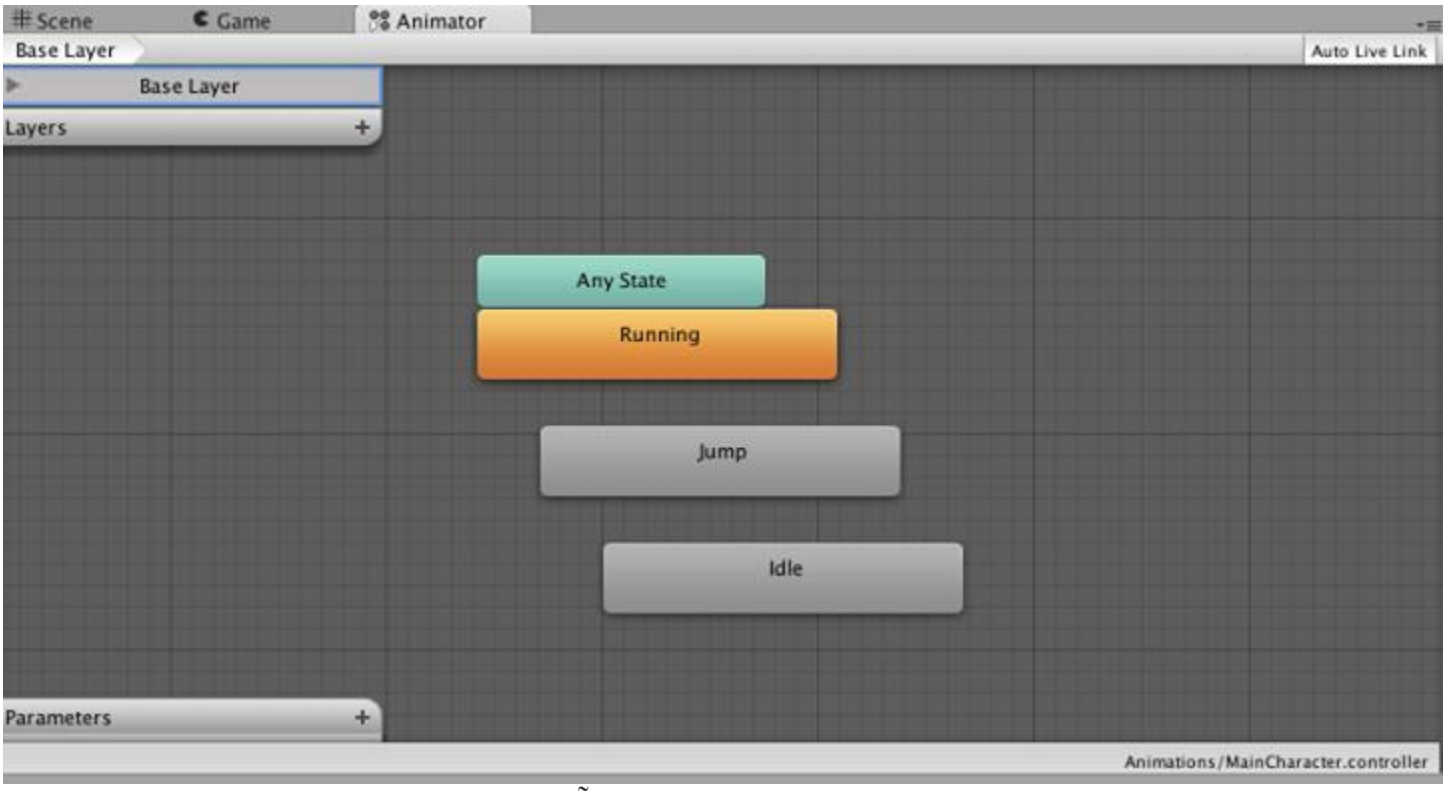
Như vậy chúng ta đã có 3 clip cho animation của đối tượng MainCharacter.

c. Điều khiển các hành động nhân vật - Animator
Ở phần trên chúng ta đã tìm hiểu cách tạo các clip hay các hành động của một animation. Với một animation như vậy ta sẽ có một Controller (MainCharacter.controller) đi kèm theo. [Hoặc nếu chưa có, chúng ta có thể add thêm và kéo thả các clip này vào :D).

Phần này ta sẽ hướng dẫn cách chuyển qua lại giữa các hành động bằng máy trạng thái.

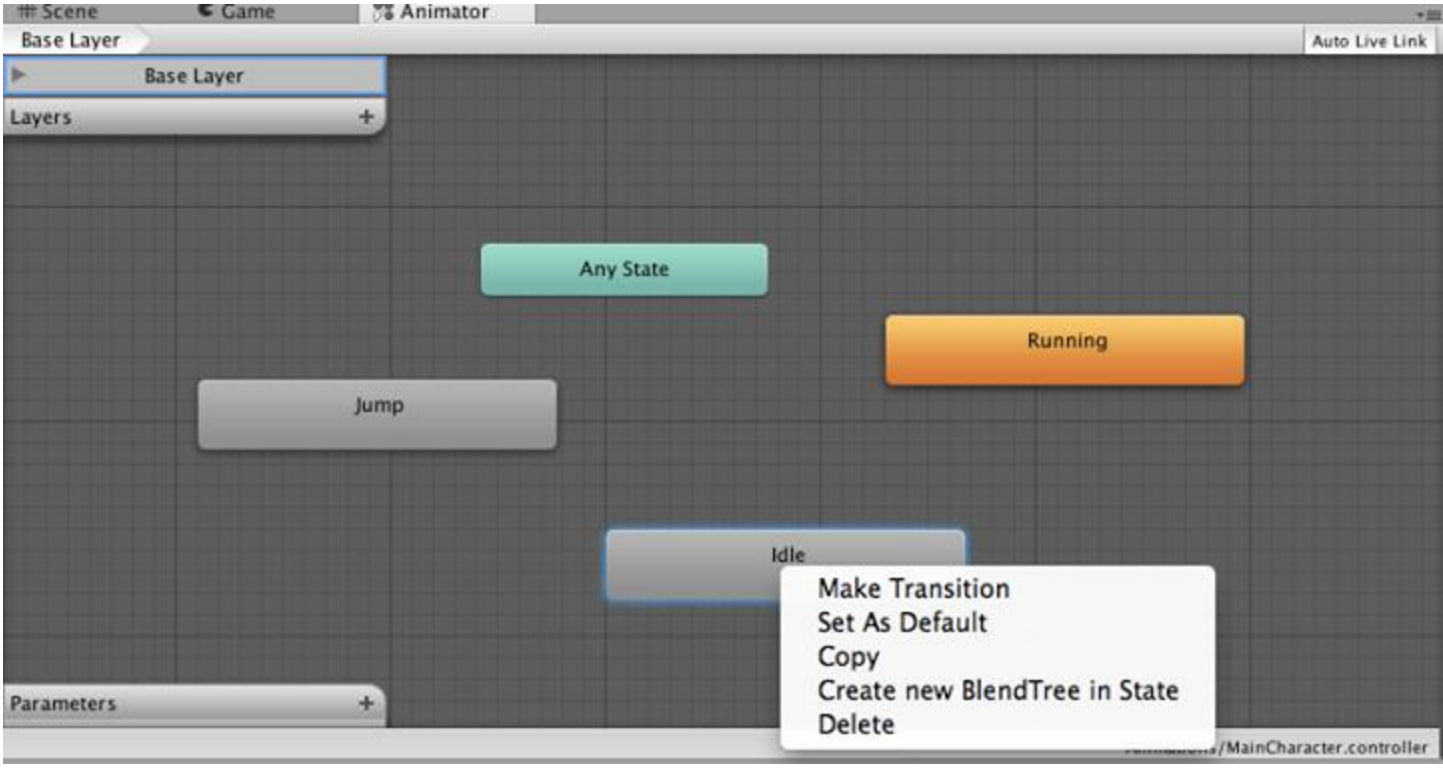


Ở cửa sổ Hierarchy chọn đối tượng MainCharacter, chọn Menu, chọn Window, chọn Animator, cửa sổ Animator sẽ xuất hiện như sau:
(Hoặc chúng ta có thể click đúp vào MainCharacter.controller cũng có kết quả tương tự.)

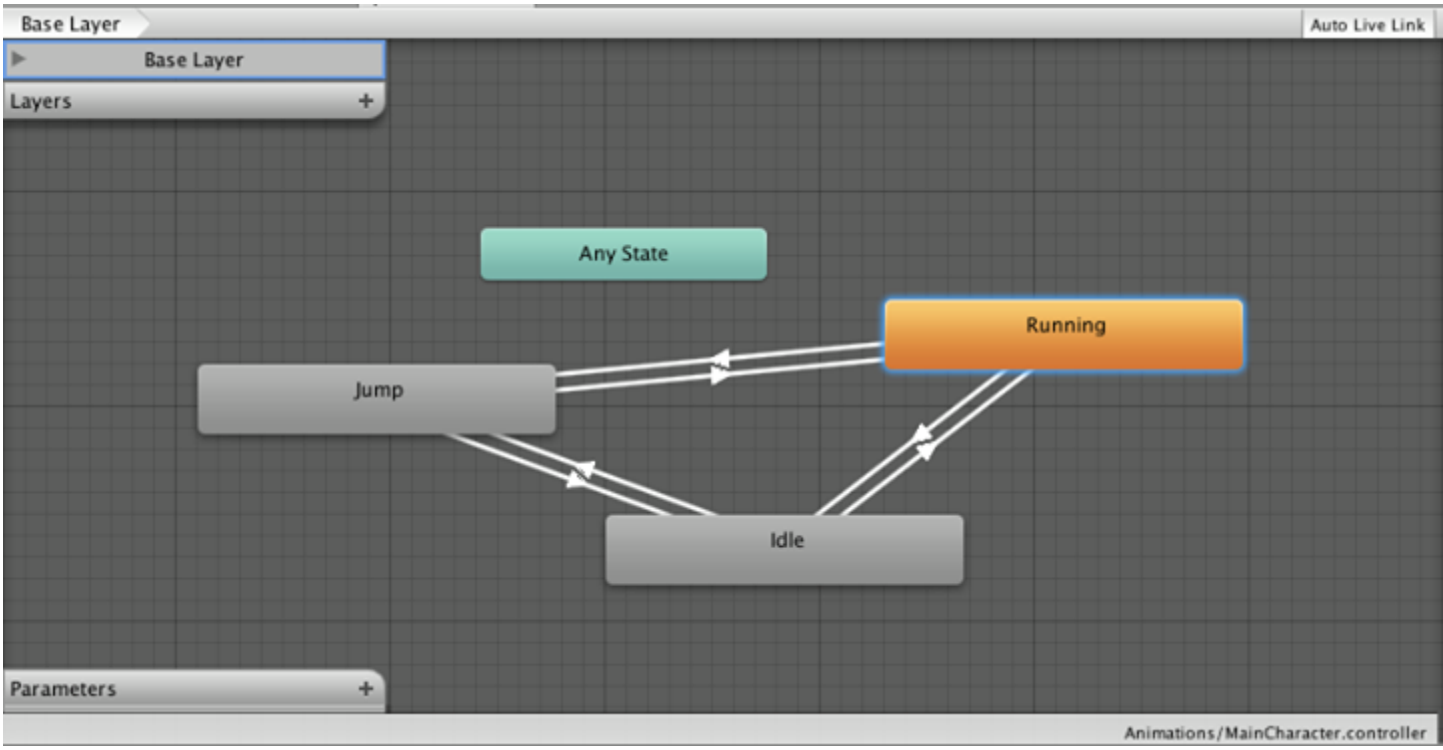


Danh sách các state, mỗi state tương ứng với một clip

Click chuột phải vào state Idle, chọn Set Default để thiết lập state mặc định cho đối tượng.



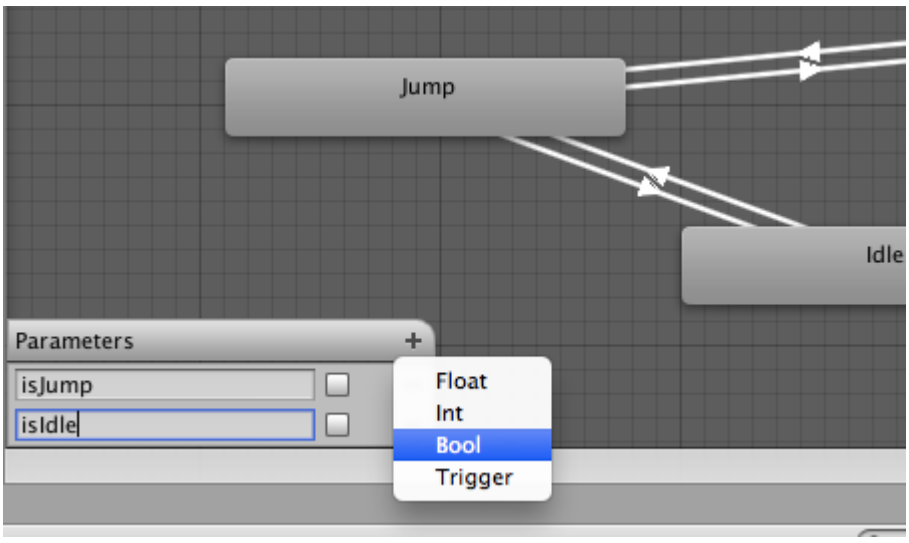
Chọn Make Transition, sau đó đưa chuột đến trạng thái đích. Với mỗi transition vừa tạo, có nghĩa rằng nhân vật từ trạng thái hiện tại có thể chuyển đổi trực tiếp qua trạng thái đích. Kết quả thu được ta gọi là máy trạng thái hay sơ đồ chuyển đổi trạng thái.



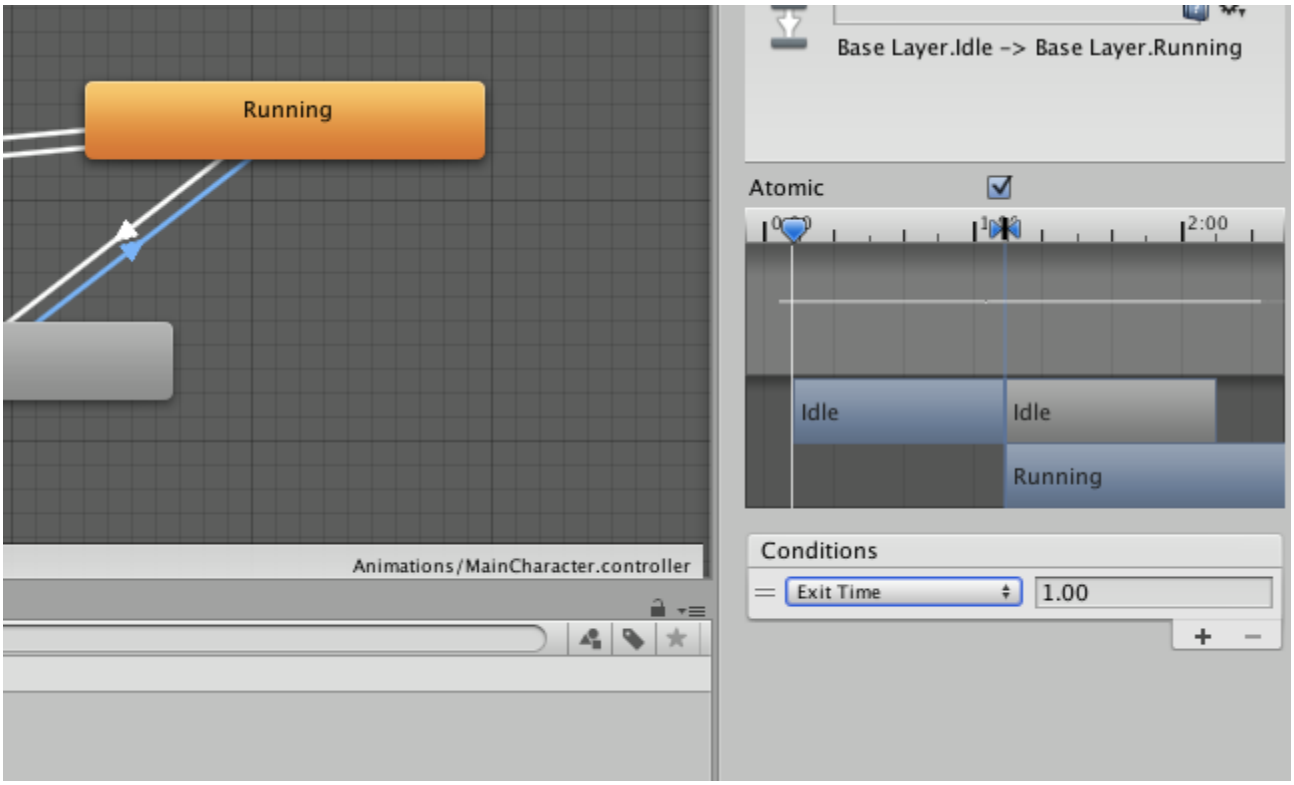
Khi đối tượng được load lên, trạng thái mặc định sẽ được thiết lập, vậy lúc nào thì sẽ chuyển qua trạng thái khác ?? Chúng ta sẽ tạo thêm các tham số, và dựa vào giá trị các tham số này để chuyển đổi các trạng thái.

Chú ý: ở đây để cho đơn giản, ta thiết lập 3 trạng thái có thể chuyển qua lại trực tiếp với nhau.

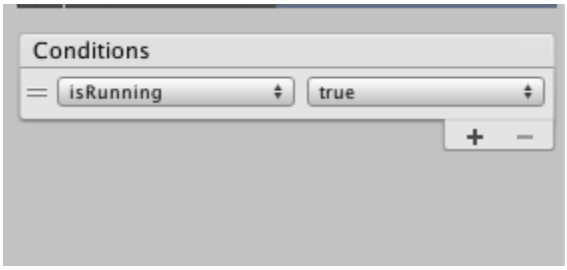
Ta sẽ tạo ra 3 tham số kiểu bool là isJump, isIdle, isRunning để điều khiển.



Để thiết lập điều kiện cho một transition, ta click chọn transition đó (transition được chọn chuyển qua màu xanh), ở cửa sổ Inspector, mục thuộc tính Conditions (điều kiện) ta sẽ thiết lập giá trị của các tham số, để xác định lúc nào thì chuyển trạng thái.



Ở đây, chúng ta chọn isRunning = true.

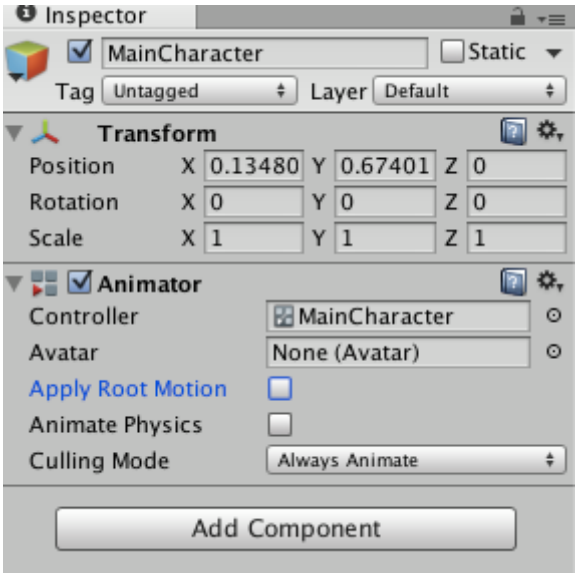


Ta có thể thêm các điều kiện khác (trường hợp chuyển đổi phụ thuộc nhiều điều kiện) bằng cách nhấn dấu cộng hoặc dấu trừ để bỏ bớt một điều kiện.

Vậy mỗi khi ta set giá trị tham số isRunning = true, trạng thái nhân vật sẽ chuyển sang Running. Tương tự ta thiết lập isIdle, isJump cho các transition khác.

Chú ý: Phần Script ta sẽ đề cập rõ, cách thiết lập giá trị các tham số này.

Cuối cùng, ở cửa sổ Hierarchy, ta chọn MainCharacter, ở cửa sổ Inspector, mục Animator, ta bỏ chọn Apply Root Motion.



Đến đây, chúng ta đã biết cách xây dựng một animation cho các đối tượng trong game và điều khiển qua lại các đối tượng đó. Ở phần tiếp theo mình sẽ hướng dẫn cho các bạn biết cách tạo và sử dụng Prefabs và Script và một số xử lý cơ bản.

Link phần tiếp theo: https://www.facebook.com/notes/hội-lập-trình-viên-game-đà-nẵng/làm-game-2d-bằng-unity-phần-4-prefab-script-và-một-số-xử-lý-cơ-bản/242866329218970