

# Bài 3

## Hoạt cảnh - Animation

Đỗ Văn Đức

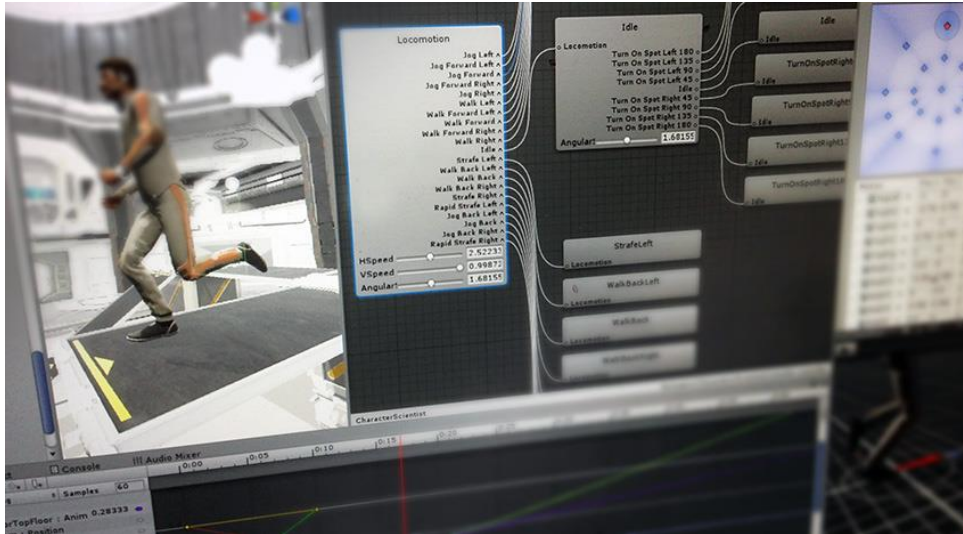
Khoa Công nghệ thông tin

Đại học Giao thông vận tải

### Nội dung

1. Khái niệm Animation
2. Kỹ thuật tạo Animation
3. Các bước tạo 1 clip Animation

# 1. Khái niệm Animation

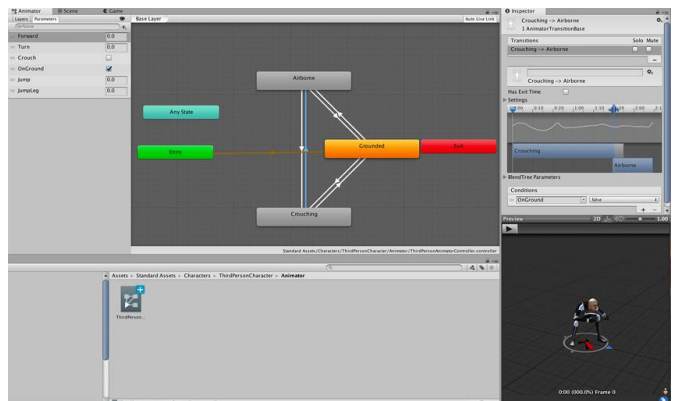


# 1. Khái niệm Animation

- 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):

- Kỹ thuật **key frame**.
- Kỹ thuật **skeletal** hay **spine**.



## 2. Kỹ thuật tạo Animation

### 2.1 Kỹ thuật key frame

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



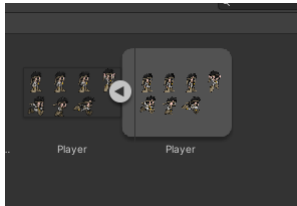
## 2. Kỹ thuật tạo Animation

### 2.2 Kỹ thuật skeletal hay spine hay bộ xương

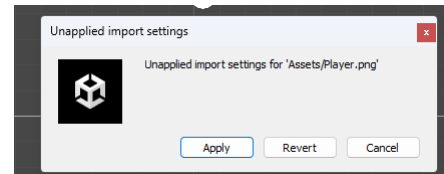
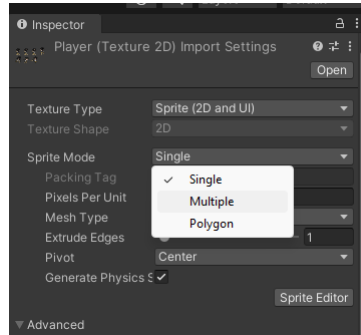
- Đối với kỹ thuật này, người ta chia đối tượng ra thành nhiều sprite, mỗi sprite là một bộ phận của đối tượng (giống như 1 khúc xương của bộ xương).
- Để tạo ra một key frame mới, ta sẽ thay đổi các sprite về vị trí, độ lớn, xoay của các sprite thành phần có liên quan đến chuyển động.
- Sau đó kết hợp các key frame lại với nhau như kỹ thuật key frame để tạo thành các animation.
- Cách này có vẻ tốn thời gian hơn, nhưng lại rất là hiệu quả, đặc biệt là tiết kiệm được nhiều bộ nhớ.



### 3. Các bước tạo 1 Clip Animation



Original sprite

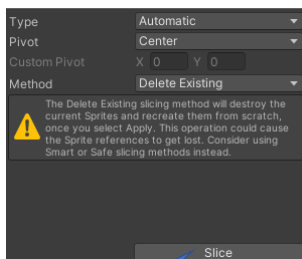


Bấm Apply để áp dụng

Thay đổi Sprite Mode từ **single** sang **multiple**  
sau đó click **Sprite Editor**

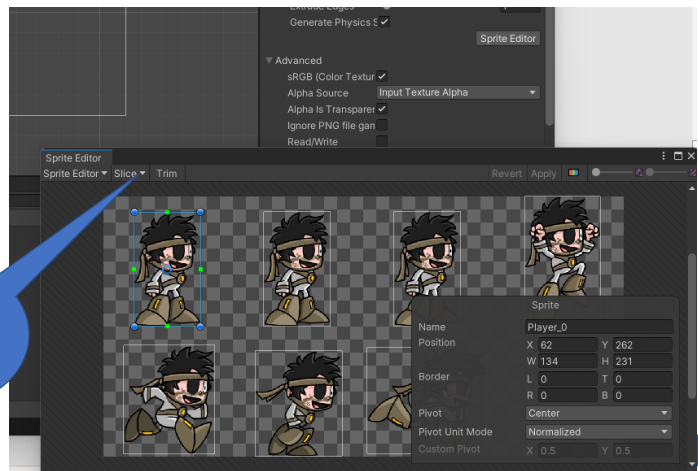
### 3. Các bước tạo 1 Clip Animation

➤ Bấm **Sprite Editor** để mở ra hộp thoại:



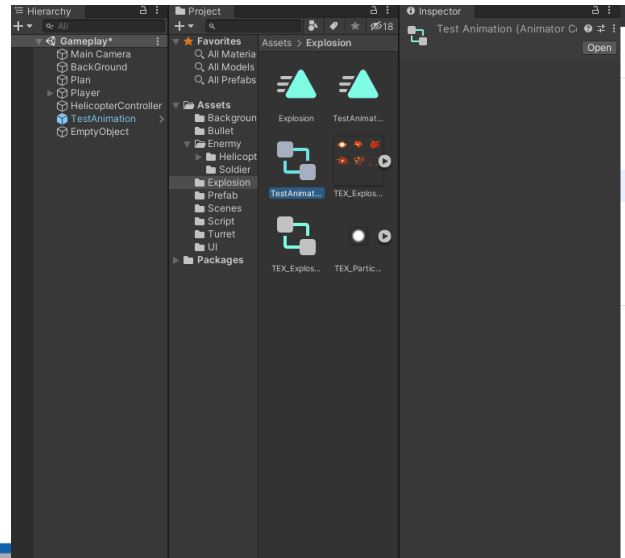
Bấm  
chọn  
Slice

Bấm  
chọn  
Slice



### 3. Các bước tạo 1 Clip Animation

➤ Bấm **Sprite Editor** để mở ra hộp thoại:



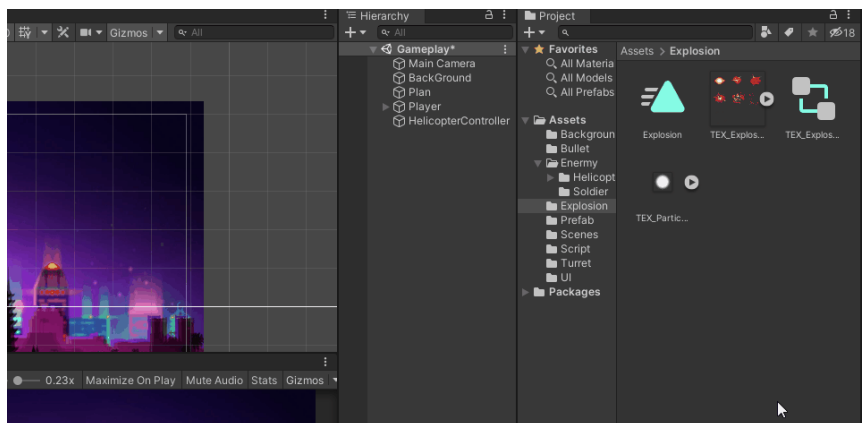
### 3. Các bước tạo 1 Clip Animation

Tạo **animation clip** và **animation controller** từ **multiple sprite** vừa tạo ra ở bước trước

Có 2 cách để tạo:

- **Cách 1:** Chọn tất cả các Sprite con từ Multiple Sprite sau đó nhấn và thả vào cửa sổ scene

Mặc định, Khi kéo thả như vậy, Unity sẽ thêm **Animator Component** vào **GameObject**, và cũng tự động tạo ra **Animator Controller** và gán vào thuộc tính **Controller** của **Animator Component**.

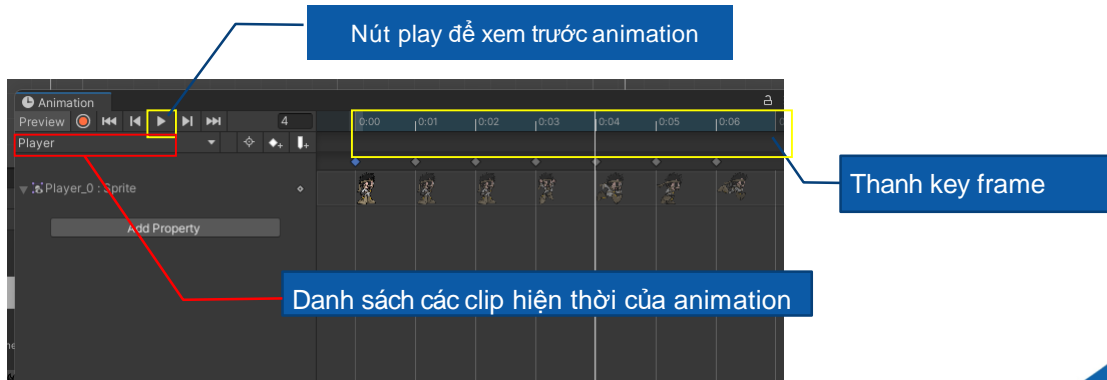


### 3. Các bước tạo 1 Clip Animation

Tạo **animation clip** và **animator controller** từ **multiple sprite** vừa tạo ra ở bước trước

Có 2 cách để tạo:

- **Cách 1:** Chọn tất cả các Sprite con từ Multiple Sprite sau đó nhấn và thả vào cửa sổ scene

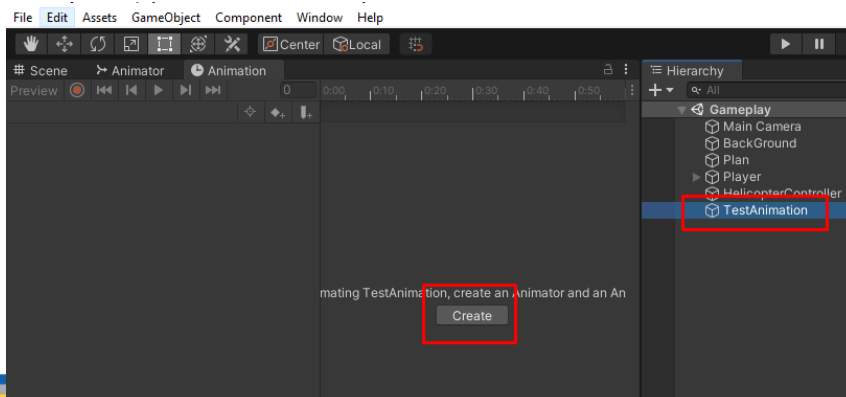


### 3. Các bước tạo 1 Clip Animation

Tạo **animation clip** và **animator controller** từ **multiple sprite** vừa tạo ra ở bước trước

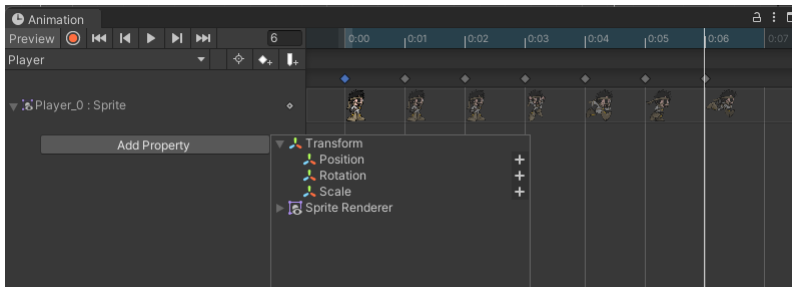
Có 2 cách để tạo:

- **Cách 2:** Tạo một **Empty GameObject**. Vào Window\Animation\ Animation (Ctrl + 6). Bấm Create để tiến hành tạo 1 Animator. Chọn tất cả các Sprite con từ Multiple Sprite sau đó nhấn và thả vào cửa sổ Animation.



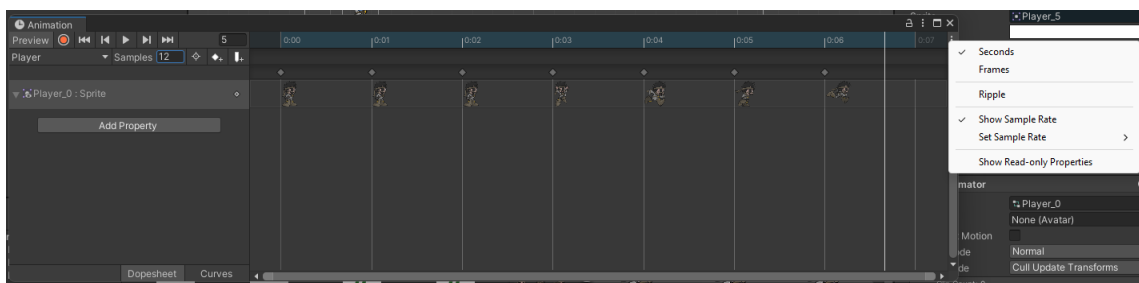
### 3. Các bước tạo 1 Clip Animation

- 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:



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

### 4. Thay đổi tốc độ Animation

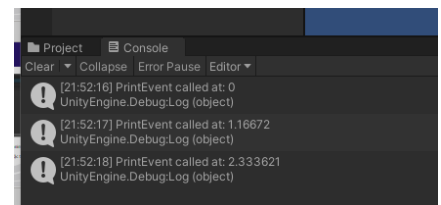
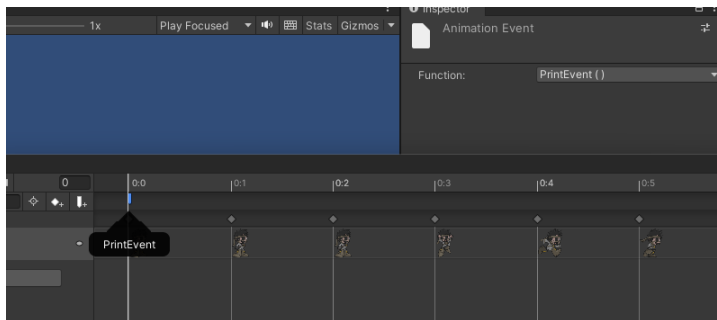


- Sample rate:** Tần số Frame/Second. Số này càng cao thì chuyển động animation càng nhanh và ngược lại
- Để thay đổi tốc độ Animation, ta có thể thay đổi giá trị Sample rate hoặc thay đổi khoảng cách giữa các frame.

## 5. Animation Event

Sử dụng **Animation Event** để gọi các hàm tại các điểm cụ thể trong các key frame. Các hàm này có thể nằm trong bất kỳ tập lệnh nào được đính kèm với GameObject.

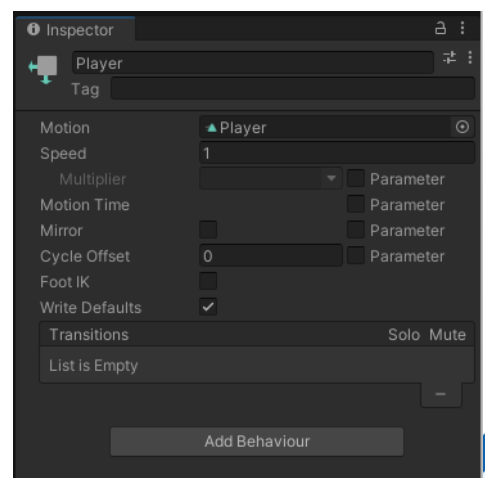
```
public void PrintEvent()
{
    Debug.Log("PrintEvent called at: " + Time.time);
}
```



## 6. Animator state

- **Animation States** là các khối xây dựng cơ bản của Animation State Machine. Mỗi trạng thái chứa một chuỗi hoạt hình được chạy khi nhân vật ở trạng thái đó.
- **Speed**: Tốc độ mặc định của Motion (**animation clip**) là 1.
- **Transitions**: Danh sách các trạng thái chuyển đổi từ trạng thái hiện tại.

Trạng thái mặc định, hiển thị màu nâu, là trạng thái sẽ được hoạt động khi Animator được kích hoạt lần đầu tiên. Để thay đổi trạng thái mặc định, chọn 1 trạng thái khác và bấm **Set As Layer Default State**

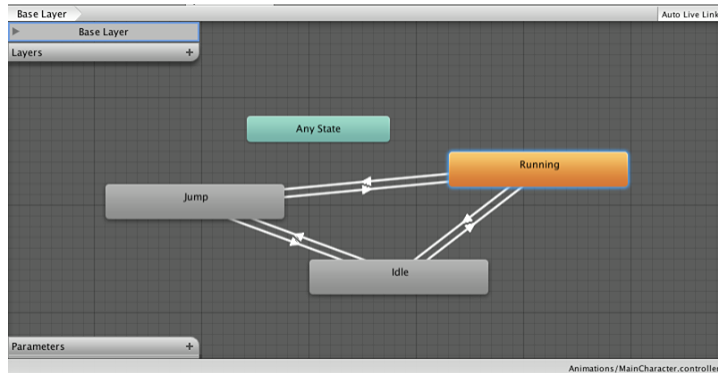




## 7. Animator state

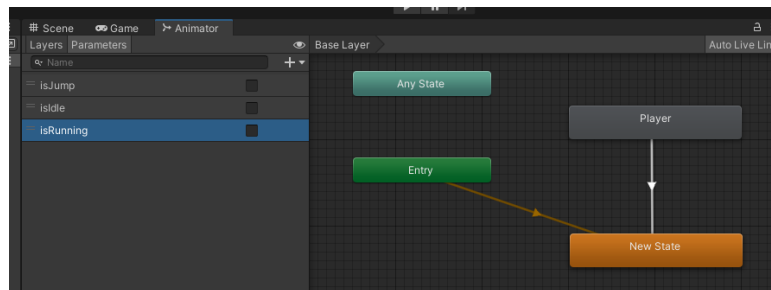
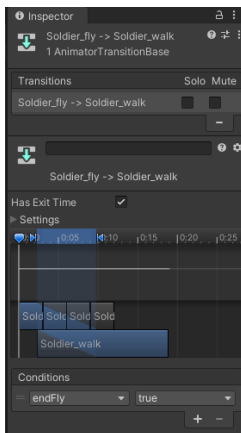
Khi đối tượng được load lên, trạng thái mặc định sẽ được thiết lập, vậy lúc nào thì sẽ chuyển qua trạng thái khác ?

Chúng ta sẽ tạo thêm các tham số, và dựa vào giá trị các tham số này để chuyển đổi các trạng thái.



## 7. Animator state

**Parameters** là các biến được định nghĩa bên trong Animator Controller mà có thể được truy xuất và thay đổi giá trị từ một scripts. Nó cho phép thay đổi trạng thái Animator bằng việc thay đổi các giá trị **Parameters**.



```
if (!endFly)
{
    endFly = true;
    anim.SetBool("endFly", endFly);
}
```

## Bài tập 1

Tạo Animation với các trạng thái Running, Idle cho Player dưới.  
Thực hiện chuyển các trạng thái theo các sự kiện nào đó.



Trạng thái Running:



Trạng thái Idle:

## Bài tập 2

Tạo Animation với các trạng thái Running, Idle cho Player dưới.  
Thực hiện chuyển các trạng thái theo các sự kiện nào đó.



## Bài tập 3

<https://assetstore.unity.com/packages/2d/characters/knight-sprite-sheet-free-93897>

