



Lập trình iOS

Bài 5. *Core Animation*

Ngành Mạng & Thiết bị di động





Nội dung

1. Giới thiệu về Animation
2. UIView Animation
3. Core Animation



1 Giới thiệu về Animation

- ❑ Animation trong mobile là một tính năng rất quan trọng, tùy theo góc nhìn người ta sẽ có các ý nghĩa khác nhau về animation.
- ❑ Dưới đây là hình minh họa về animation đơn giản trong một ứng dụng chạy trên nền iOS:





Nội dung

1. Giới thiệu về Animation

2. UIView Animation

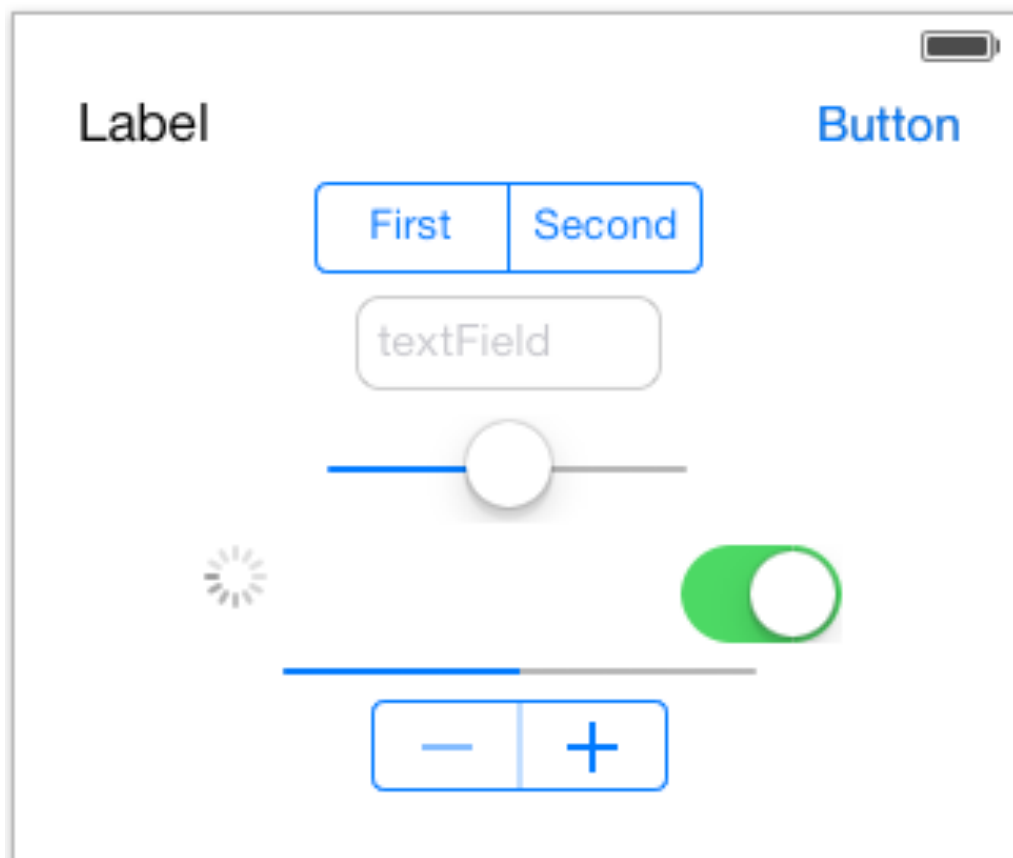
- UIView trong IOS
- Các thuộc tính của UIView
- Animation trong UIViews
- Một số ví dụ về Animation trong UIView

3. Core Animation



2.1 UIView trong IOS

- ❑ UIView là một lớp cha cho các object được thể hiện trên màn hình trong iPhone (ở đây không bao gồm Core Animation và OpenGL). Button, text fields, images... đều là các đối tượng con của UIView.





2.1 UIView trong IOS

- ❑ Một UIView có thể chứa các view khác. Ta có thể coi UIView như một superview dùng để chứa các view khác và việc thêm hoặc xóa các view vào superview tương đối đơn giản như sau:

```
// Thêm một view  
[viewObj addSubview: otherViewObj];  
// Xóa một view  
[otherViewObj removeFromSuperview];
```



2.2 Các thuộc tính của UIView

- ❑ Bằng việc thay đổi và kết hợp thay đổi các thuộc tính size, location, rotation hoặc transparency ta có thể tạo ra các animation dựa trên các tính năng được cung cấp sẵn của UIView.

- ❑ Các thuộc tính cơ bản thường được sử dụng để tạo ra các animation:
 - **frame**: Di chuyển, thay đổi kích thước của view.
 - **center**: Di chuyển view.
 - **bounds**: Thay đổi khung bao của view.
 - **transform**: Rotate, scale, translate, skew.
 - **alpha**: Thay đổi mức độ transparency trong đoạn $[0,1]$.
 - **backgroundColor**: Thay đổi màu nền.



2.3 Animation trong Views

- ❑ Thao tác hàm cơ bản để gọi thực hiện animation trong UIView:

```
[UIView animateWithDuration: [NSTimeInterval] animations:^(  
    // các hành động  
)];
```

- ❑ Ngoài ra ta còn có phương thức với trigger nhận sự kiện completion sau khi animation được thực hiện như sau:

```
[UIView animateWithDuration: [NSTimeInterval] animations:^(  
    // Các hành động  
} completion:^(BOOL finished) {  
    // Các hành động sau khi thực hiện xong  
}];
```




2.4 Một số ví dụ về Animation trong UIView

❑ Dịch chuyển (Move):

```
CGRect newFrame = viewObj.frame;
newFrame.origin.x += 500;
// dịch chuyển sang phải 500pts
[UIView animateWithDuration:1.0 animations:^(
    viewObj.frame = newFrame;
)];
```

❑ Xoay (Rotation)

```
float degrees = 45.0;
float radians = (degrees/180.0) * M_PI;
[UIView animateWithDuration:0.25 animations:^(
    viewObj.transform = CGAffineTransformMakeRotation(radians);
)];
```



2.4 Một số ví dụ về Animation trong UIView

❑ Xoay tới và ngược (Undoing Rotations):

```
viewObj.transform = CGAffineTransformIdentity;
float degrees = 45.0;
float radians = (degrees/180.0) * M_PI;
[UIView animateWithDuration:0.25 animations:^(
    viewObj.transform = CGAffineTransformMakeRotation(radians);
} completion:^(BOOL finished){
    [UIView animateWithDuration:0.25 animations:^(
        viewObj.transform = CGAffineTransformIdentity;
    )];
}];
```



2.4 Một số ví dụ về Animation trong UIView

❑ Hiệu ứng lồng:

```
CGRect newFrame = viewObj.frame;
newFrame.origin.x += 500;
// dịch chuyển sang phải 500pts
[UIView animateWithDuration:1.0 animations:^(
    viewObj.frame = newFrame;
)completion:^(BOOL finished){
    // code sẽ thực hiện sau khi animation
    // di chuyển view hoàn thành
    // trường hợp này, ta thực hiện một
    // animation khác
    [UIView animateWithDuration:1.0 animations:^(
        viewObj.alpha = 0.0;    // fade out
    )completion:^(BOOL finished){
        [viewObj removeFromSuperview];
    }];
}];
```



Nội dung

1. Giới thiệu về Animation

2. UIView Animation

3. Core Animation

- Giới thiệu
- Cơ bản về Core Animation trong IOS
- Ví dụ

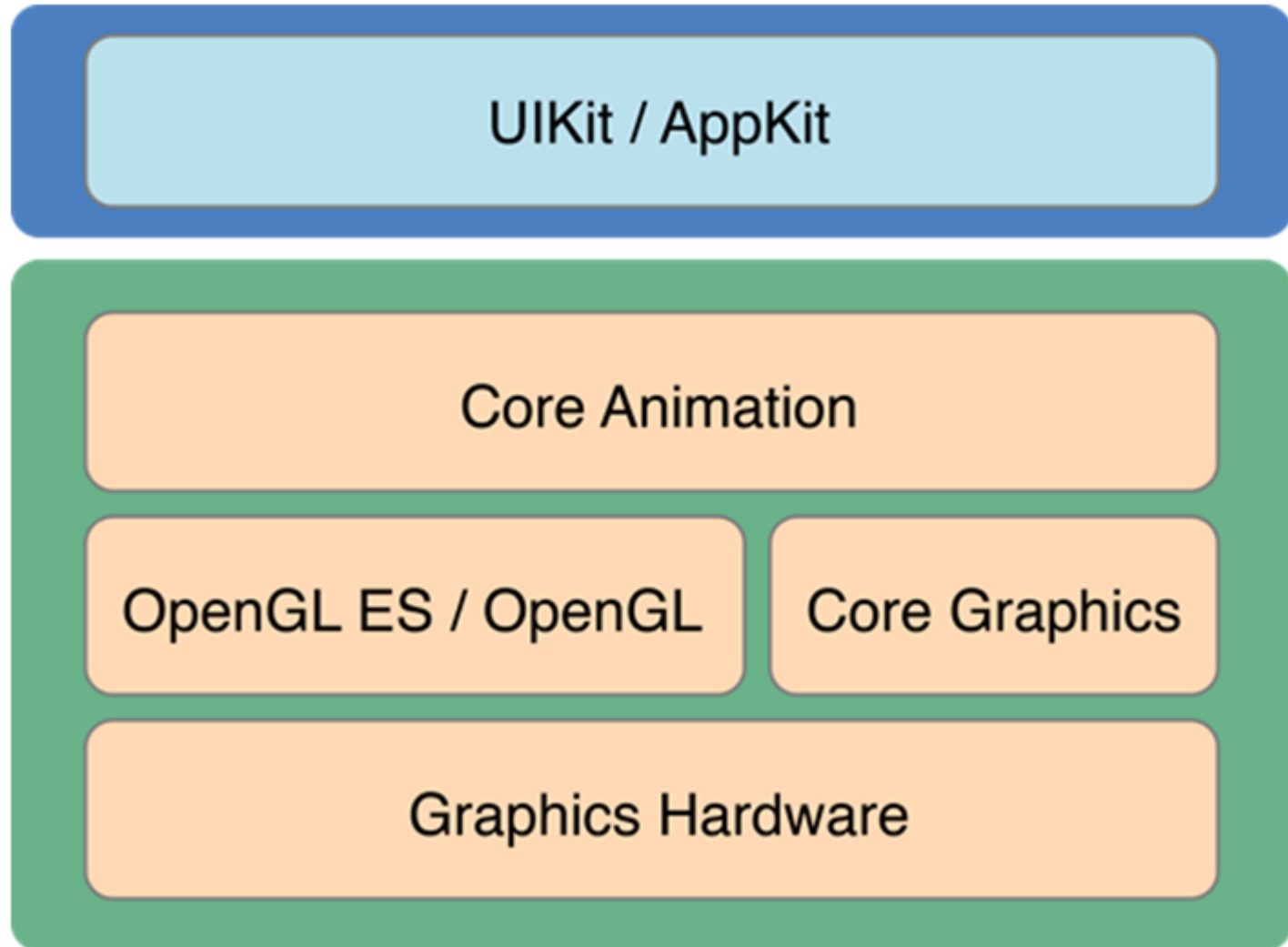


3.1 Giới thiệu

- ❑ Core Animation là một nền tảng trong việc thiết kế đồ họa và các chuyển động hình ảnh được tích hợp sẵn trong iOS và OS X.
- ❑ Với Core Animation có thể giúp ta tạo ra các hiệu ứng, chuyển hoạt hình ảnh trên ứng dụng.



3.1 Giới thiệu





3.2 Cơ bản về Core Animation trong iOS

- ❑ Layer Object
- ❑ Cách Core Animation thể hiện một đối tượng
- ❑ Các animation cơ bản trong iOS
- ❑ Các thuộc tính hình học của Layout Object

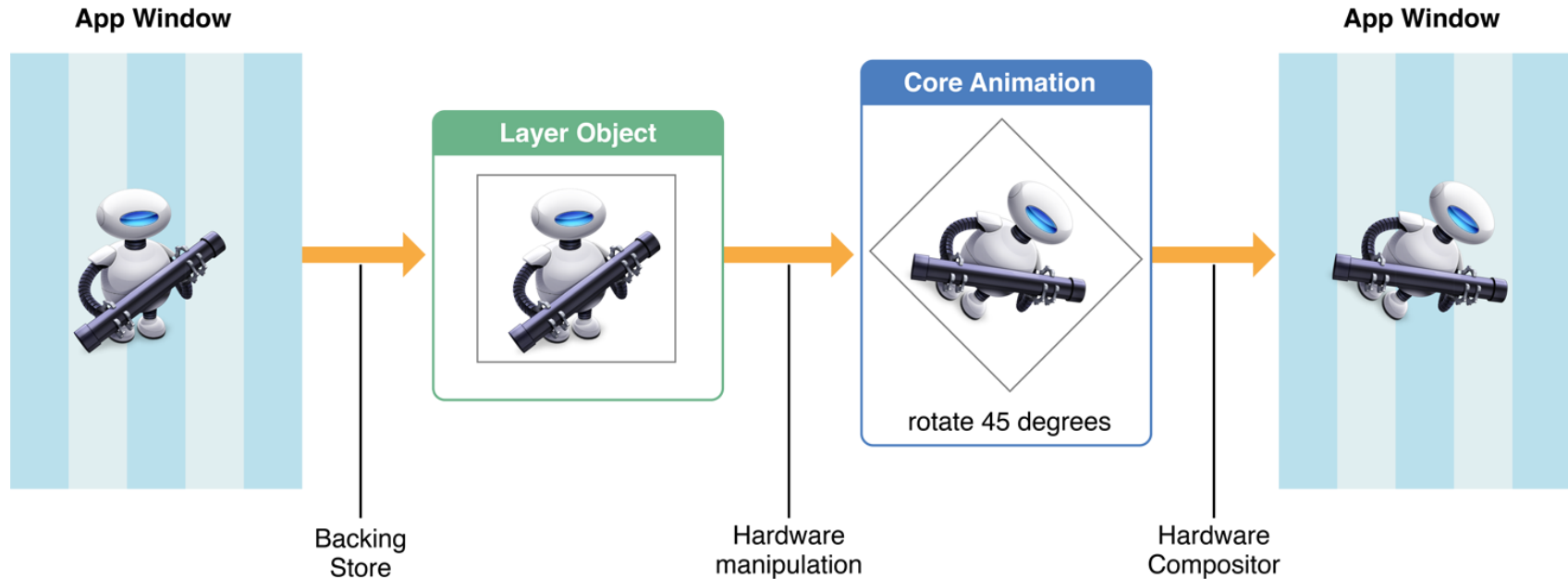


3.2.1 Layer Object

- ❑ Layer Object là một mặt phẳng 2 chiều được vẽ trong không gian (3D) và đây là một đối tượng cực kỳ quan trọng khi làm việc với Core Animation.



3.2.2 Cách Core Animation thể hiện một đối tượng

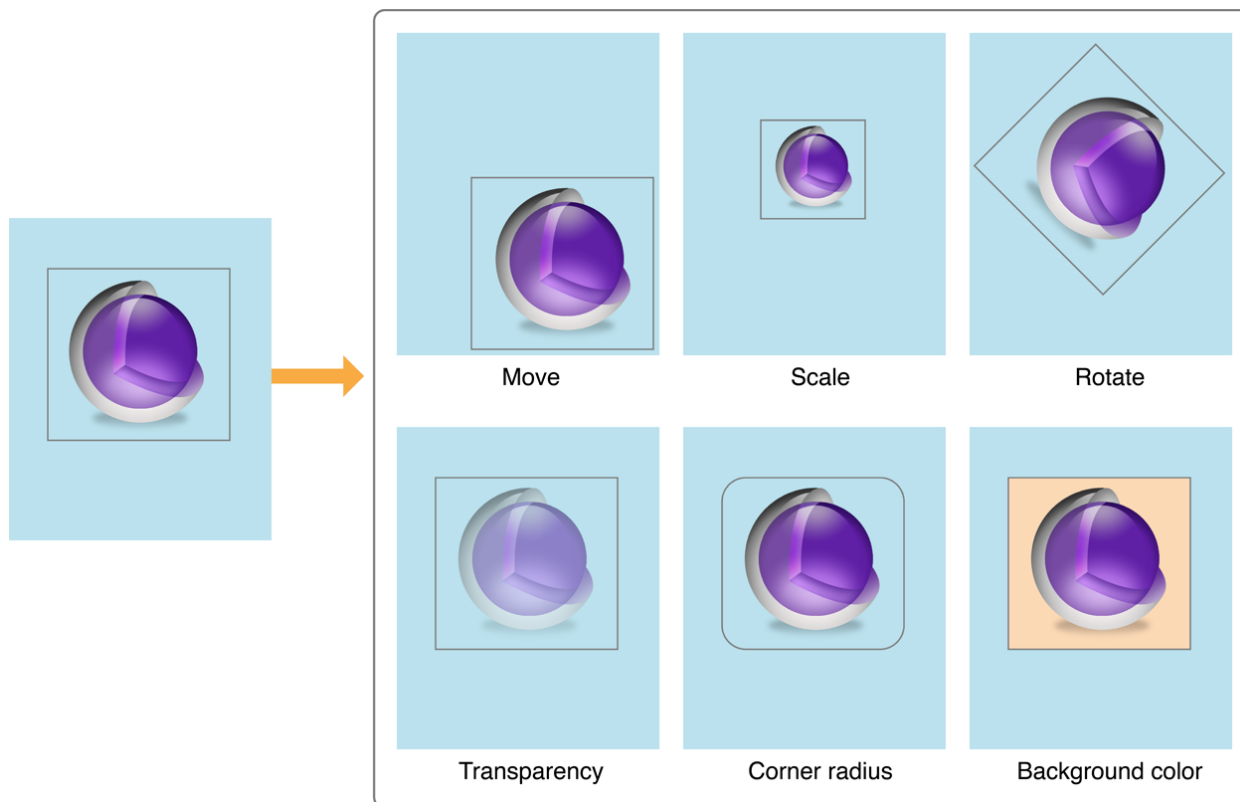


- **Backing Store:** Một layout muốn thể hiện một content lên màn hình thì nó sẽ thực hiện việc “capture” và “cache” đối tượng cần thể hiện vào bitmap.
- **Hardware manipulation:** Quá trình thay đổi các thông tin, thuộc tính của layout tạo nên sự chuyển hoạt của đối tượng.
- **Hardware Compositor:** Tiếp nhận các thay đổi và chuyển các thông tin mới xuống phần cứng đồ họa.



3.2.3 Các animation cơ bản trong iOS

- ❑ Để thể hiện được các thao tác chuyển hoạt, bản thân Core Animation sẽ thực hiện vẽ từng khung ảnh (frame-by-frame) lên phần cứng đồ họa.





3.2.4 Các thuộc tính hình học của Layer Object

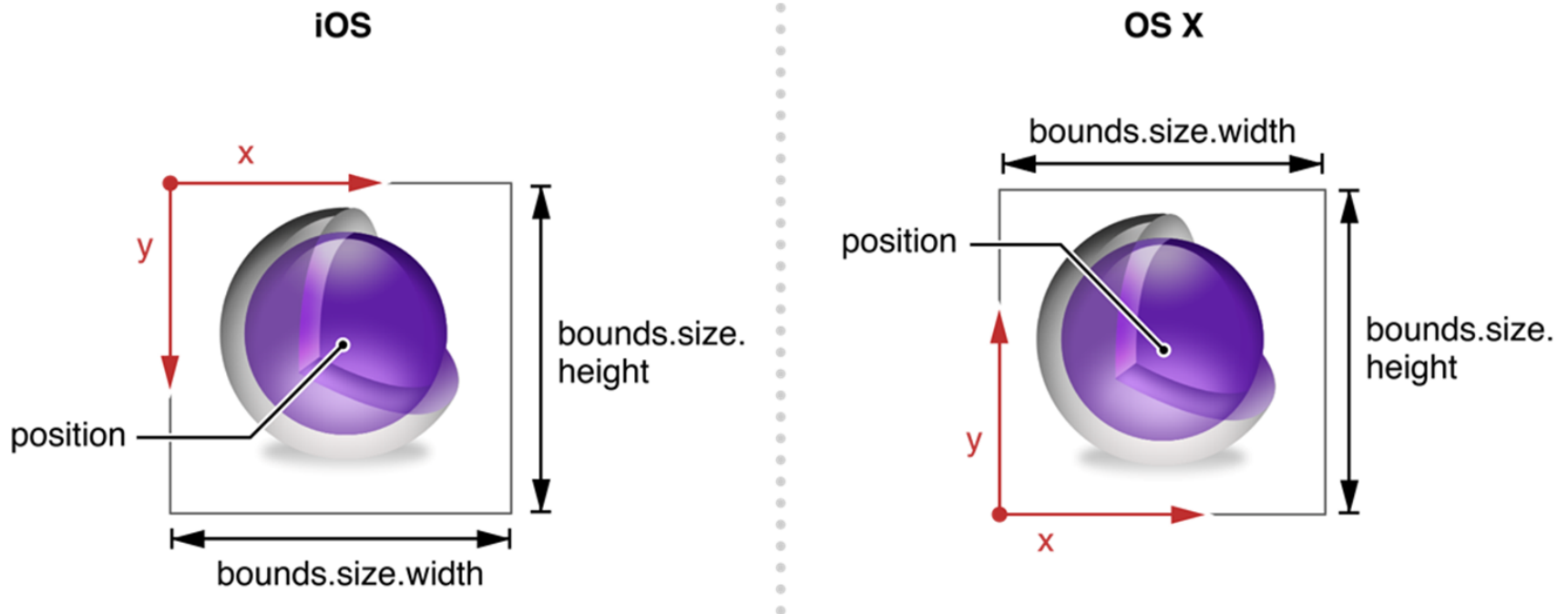
❑ Các hệ qui chiếu tọa độ:

- Point-based Coordinate systems: sử dụng cho các thuộc tính mà giá trị của nó được thể hiện trực tiếp thông qua giá trị của góc tọa độ: position, bound...
- Unit Coordinate systems: sử dụng cho các thuộc tính mà giá trị của nó được thể hiện gián tiếp thông qua một giá trị khác (không nhất thiết là góc tọa độ): anchorPoint.

3.2.4 Các thuộc tính hình học của Layer Object



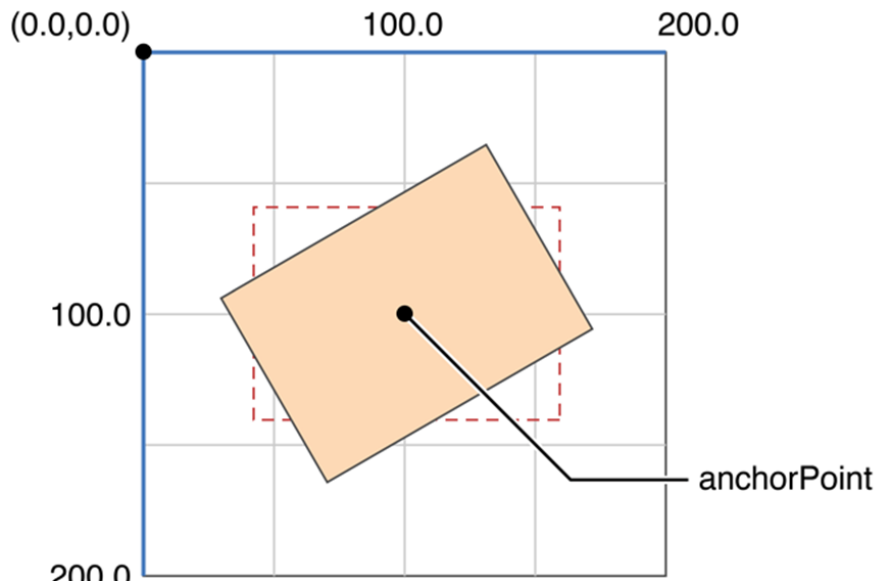
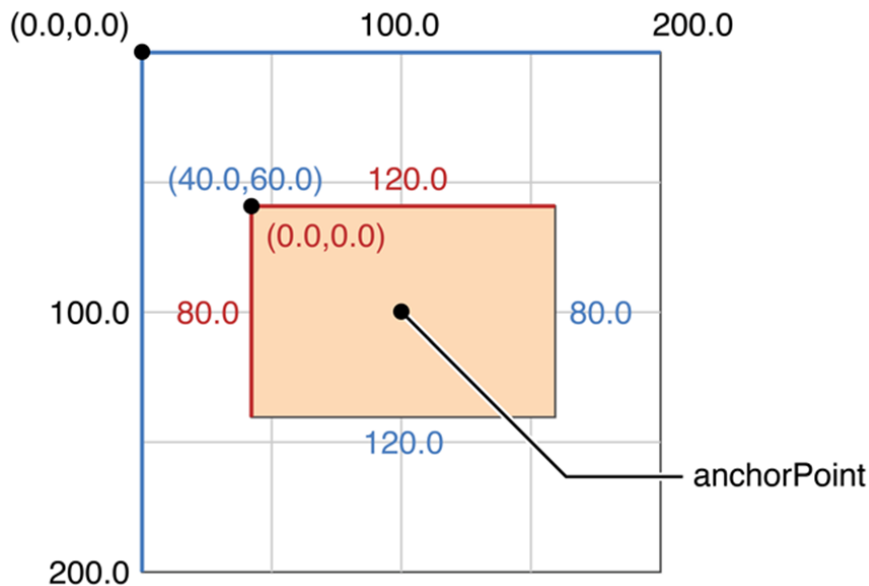
- ❑ So sánh hệ tọa độ chuẩn (hệ Point-based) trong iOS và OS X



3.2.4 Các thuộc tính hình học của Layer Object



❑ Chọn anchorPoint tại tâm ảnh

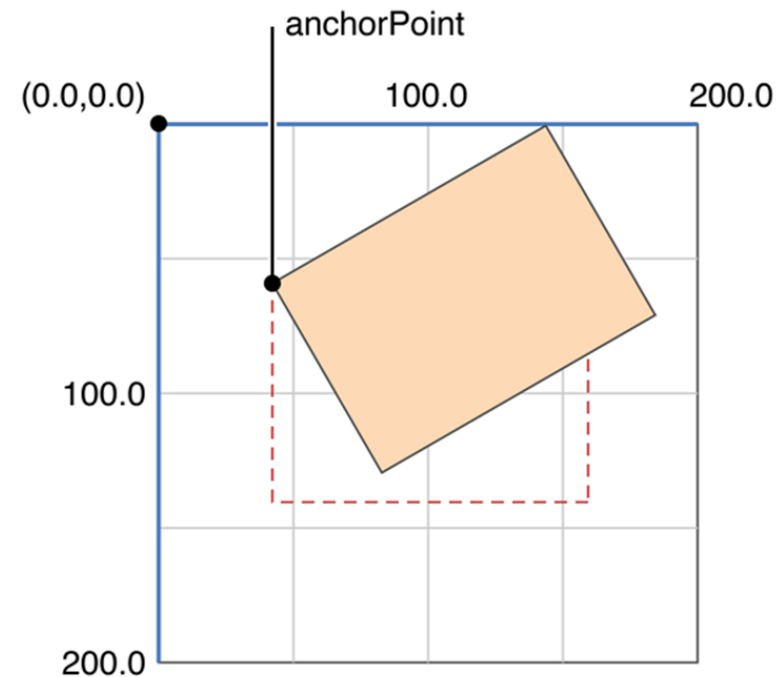
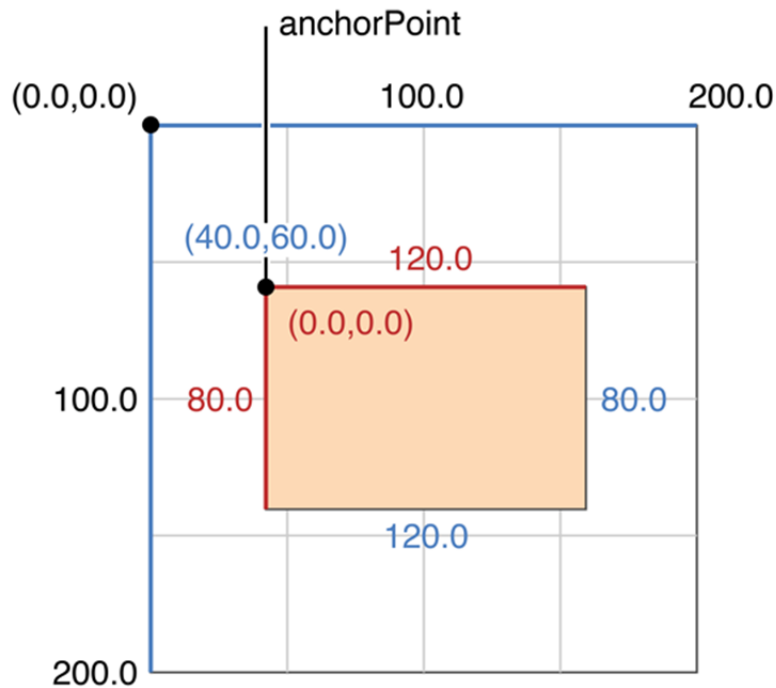


❑ Xoay ảnh 1 góc α pha

3.2.4 Các thuộc tính hình học của Layer Object



- ❑ Chọn anchorPoint tại góc trên trái của ảnh

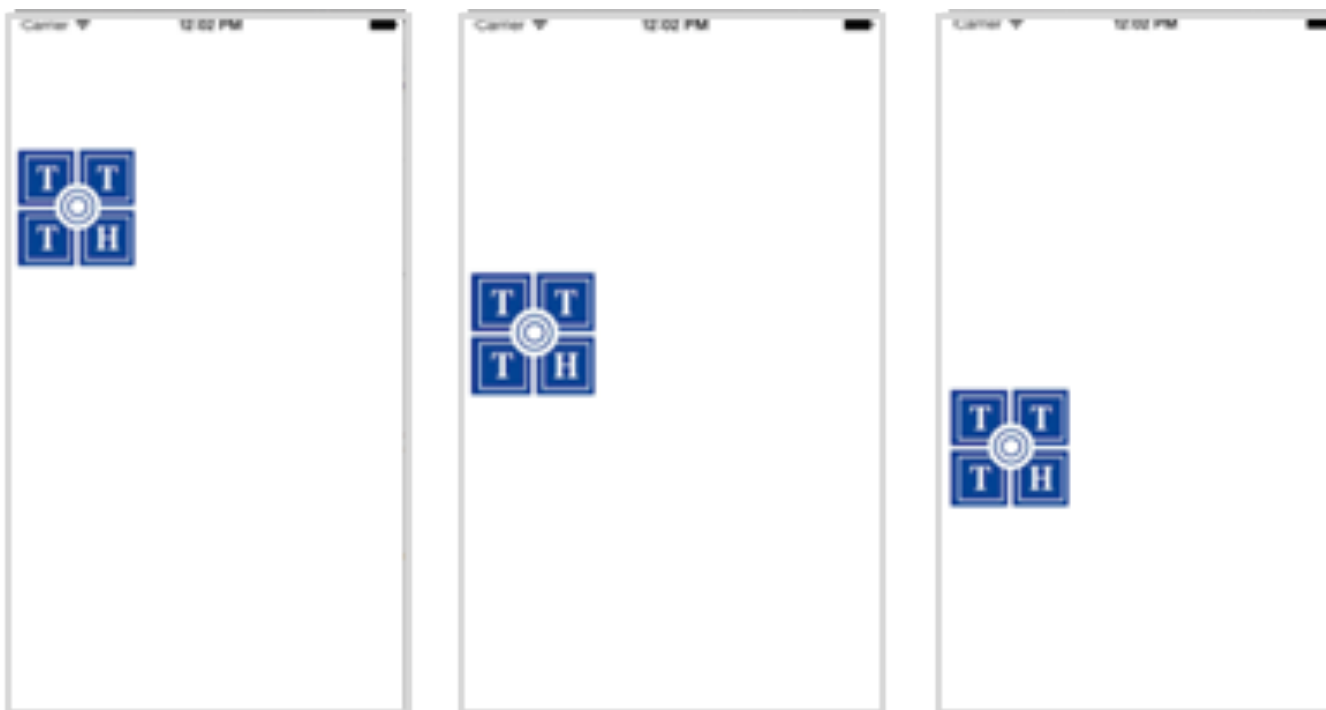


- ❑ Xoay ảnh 1 góc α pha



3.3 Ví dụ

- ❑ Xây dựng một demo cho ảnh di chuyển như hình bên dưới



Thảo luận

