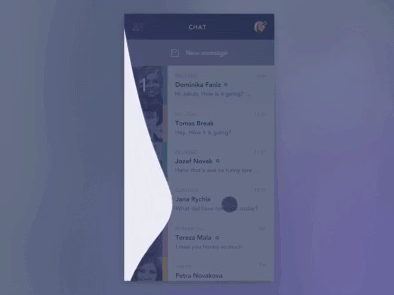
1. Animation
   1. Animation là gì?

**Animating = “Change over time”.**

*Change = Curves*

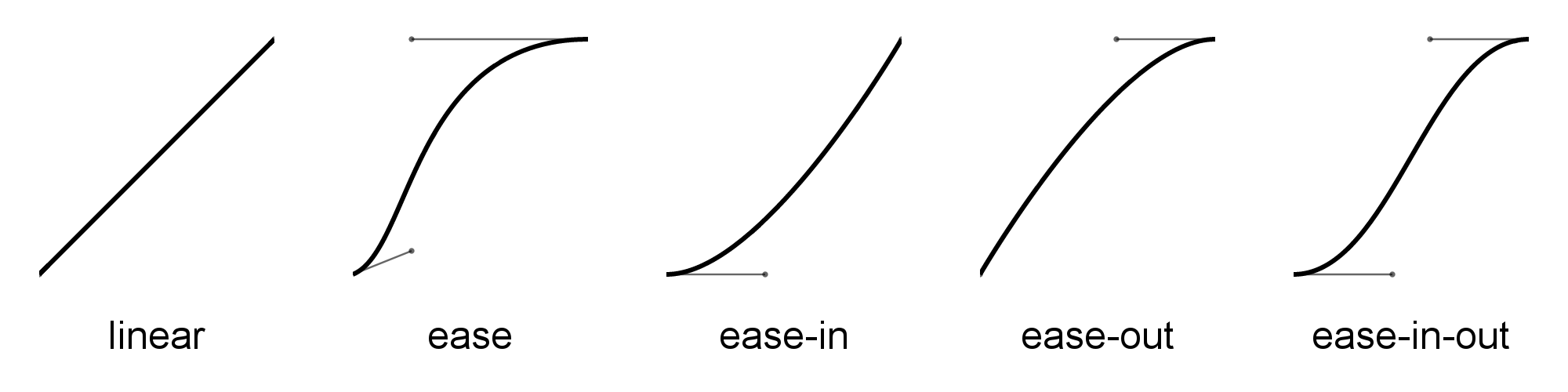
*Time = Timing*



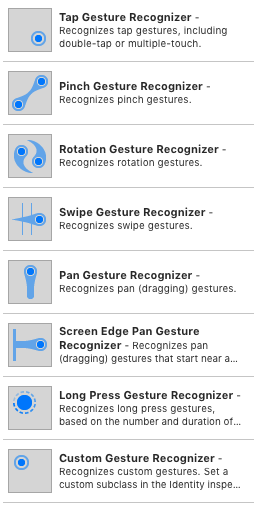
* 1. Mục đích

UI tự nhiên, dễ sử dụng, thu hút và cảm thấy thoải mái hơn khi sử dụng.

* 1. Các kiến thức cần biết
     1. Curves: Linear, Ease In/Out/In Out, Bounch, Spring,…

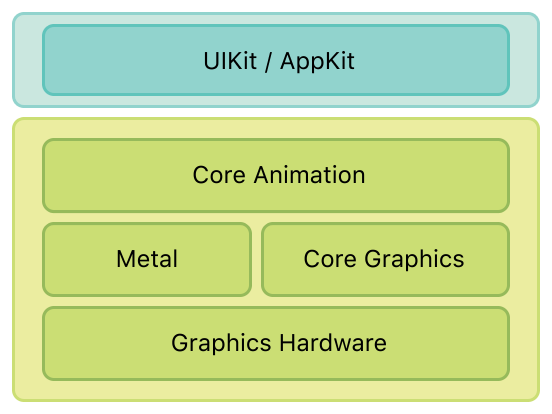


* + 1. Timing
    2. Properties: Position, Scale, Rotation, Opacity,…
    3. Events/Triggers: Tap, Pinch, Pan, Swipe,…



1. Core Animation
   1. Overview
      1. Core Animation là 1 framework animation tạo animating cho UI và các hiệu ứng.
      2. Nhẹ, high-performance
      3. Mạnh mẽ, đơn giản, dễ sử dụng

Core Animation Architecure



* 1. Components
     1. Layers
     2. Animations
     3. Transactions
     4. Transformations

1. Components
   1. Layers

**CALayer**

|  |  |
| --- | --- |
| **CALayer** | **UIView** |
| * Backing Layer * Lightweight * Rendering, animating | * Kế thừa UIResponder * Mỗi UIView đều chứa 1 CALayer (root) * Handle events |

*Ví dụ về CALayer:*

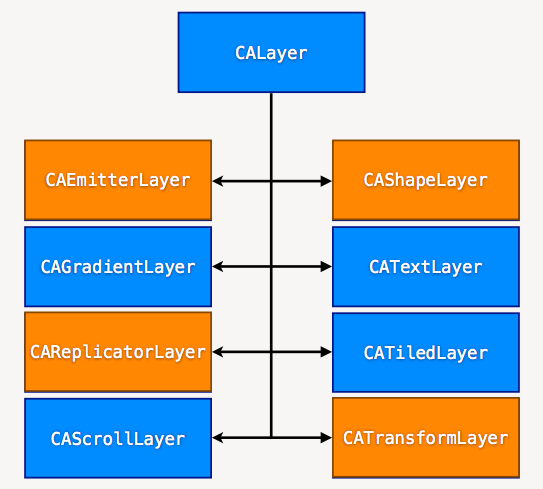
**CALayer Geometry**

|  |  |
| --- | --- |
| layer.position = CGPointMake(150, 200)  layer.bounds = CGRectMake(0, 0, 170, 220)  layer.transform =  CATransform3DMakeRotation(M\_PI / 4, 0, 0, 1) |  |
| layer.anchorPoint = CGPointMake(0.5, 1)  layer.transform =  CATransform3DMakeRotation(M\_PI / 4, 0, 0, 1) |  |

**CALayer Appearance**

|  |  |
| --- | --- |
| layer.cornerRadius = 5.0;  layer.borderWidth = 2.0;  layer.borderColor = [UIColor redColor].CGColor;  layer.shadowOpacity = 0.7;  layer.shadowRadius = 5.0;  layer.shadowOffset = CGSizeMake(1, 1);  layer.shadowColor = [UIColor blackColor].CGColor;  layer.mask = maskLayer; |  |

**CALayer Class hierachy**

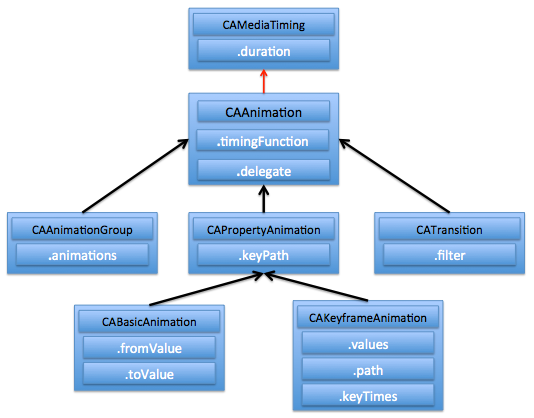


* CAEmitterLayer: dùng để tạo ra các hiệu ứng theo thời gian thực như khói, lửa, mưa ...
* CAShapeLayer: là một lớp con hiển thị các đồ họa vector thay cho ảnh bitmap.
* CAGradientLayer: dùng để tạo ra hiệu ứng chuyển màu giữa 2 hay nhiều màu.
* CATextLayer: hiển thị nội dung text.
* CAReplicatorLayer: được thiết kế nhằm tạo ra một tập hợp các lớp giống nhau một cách hữu hiệu nhất. Nó hoạt động dựa trên việc vẽ lại 1 hay nhiều bản sao của từng sublayer, áp dụng các phép biến hình lên từng bản sao.
* CATiledLayer: dùng để hiện thị 1 ảnh rất lớn bằng cách tách nhỏ thành các ảnh nhỏ hơn để hiển thị khi cần thiết
* CAScrollLayer: cho phép tạo hiệu ứng "scroll" cho các nội dung trong layer, dùng để hiển thị các nội dung lớn hơn biên (bound) của layer
* CATransformLayer: dùng để tạo các đối tượng 3D
  1. Animations

**Implicit Animations**

* Hầu hết các CA properties đều là animatable.
* Chuyển đổi từ trạng thái cũ sang trạng thái mới bằng hàm tính thời gian mặc định. (trên 1/4s)
* Sử dụng linear curve mặc định.
* Sử dụng transaction để quản lý trong class CATransaction.
* Có thể thay đổi các thuộc tính mặc định của CATransation, hoặc disable implicit animation.

**Explicit Animations**

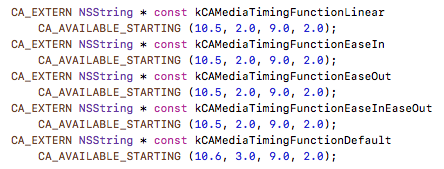


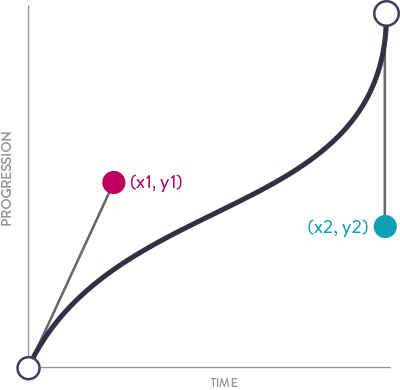
CAMediaTiming:

* *Duration*
* *repeatCount*
* *repeatDuration*
* *removedOnCompletion*: defaul is YES

CAAnimation: CAMediaTiming

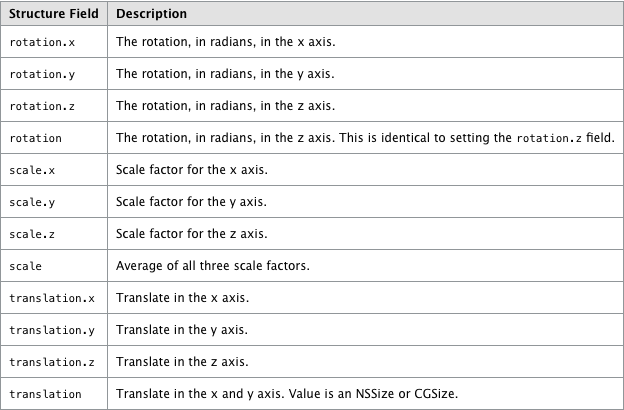
* *timingFunction*: linear, easeIn, easeOut, easeInEaseOut, default
* *delegate*
  + animationDidStart:
  + animationDidStop:finished





CAPropertyAnimation: CAAnimation

* Là 1 abstract class.
* *keyPath*: chỉ định rõ tên thuộc tính của CALayer (NSString) và chỉnh sửa giá trị đó.



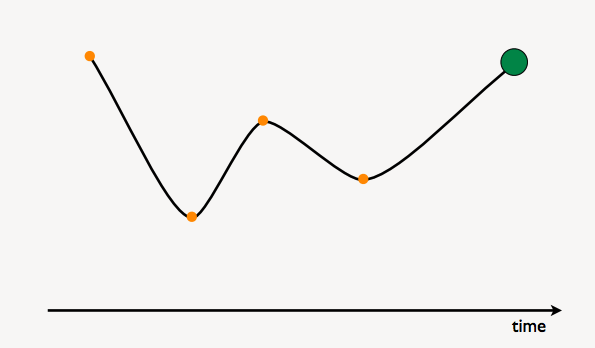
CABasicAnimation: CAPropertyAnimation

* *fromValue*: khởi tạo giá trị tương ứng với thuộc tính của keyPath
* *toValue*: giá trị kết thúc tương ứng với thuộc tính của keyPath

Trong quá trình biễu diễn animation, giá trị fromValue sẽ chuyển dần sang giá trị toValue ứng với thuộc tính của keyPath xuyên suốt **duration** được xác định ở trên.

CAKeyframeAnimation: CAPropertyAnimation

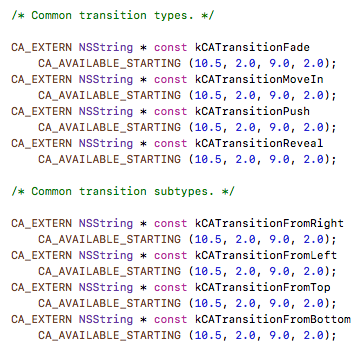
* CAKeyframeAnimation sử dụng một array để lưu nhiều giá trị fromValue và toValue
* *Values*: là mảng chứa dữ liệu ở trên. Những phần tử trong mảng gọi là “keyframes”. Animated Object sẽ hiển thị mỗi keyframe trong một khoảng thời gian.
* *Path*: CGPathRef\CGMutablePathRef – chỉ hoạt động với CALayer properties chứa CGPoint data (anchorPoint, position).
* *keyTimes*: mảng chứa thời gian bắt đầu thực thi mỗi keyframe tương ứng.



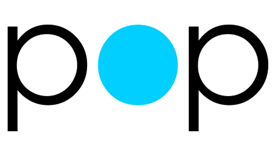
CAAnimationGroup: CAAnimation

* Được sử dụng để chạy concurrent nhiều animation.
* *Animatios*: array chứa các animated objects.
* Mặc định, các animation sẽ chạy cùng thời điểm những ta có thể thay đổi thời gian beginTime của các animated objects.

CATransition: CAAnimation

* Được dùng để tạo hiệu ứng chuyển đổi giữa các screen, tạo và hủy ViewController.
* UINavigationController sử dụng CATransition để push controller lên screen.
* Types: animation transition type
* subTypes: animation transition direction

1. Ứng dụng demo
2. Đánh giá
3. POP
   1. POP



* 1. CAAnimation vs. POP

*CAAnimation*: Khi tạo và áp dụng animation vào layer, CA kèm những parameters của nó và gởi thông qua IPC tới 1 background process là **Backboadrdd**. Còn App sẽ gởi những thông tin về mỗi layer đơn đang có trên màn hình.

Backboardd xử lý layer hierachy và renders nó bằng OpenGL (Metal). Chạy các animation được áp dụng lên layers một cách độc lập với App. Chỉ có callbacks mà App nhận được khi bắt đầu hoặc kết thúc animation. Nếu main thread bị block thì animation vẫn được tiếp tục, không gây lag.

*POP*: Sử dụng timer update, và tính toán tiến trình của animation. Sau đó áp dụng những giá trị đó vào layers đang aninate. Khi các tham số của layer được thay đổi thì app sẽ gởi tín hiệu tới backboardd yêu cầu render lại màn hình. Tức là app sẽ điều khiển quá trình animate.

|  |  |  |  |
| --- | --- | --- | --- |
| **CA** | | **POP** | |
| **Advantage** | **Disadvantage** | **Advantage** | **Disadvantage** |
| Xử lý trên GPU | Code nhiều hơn | APIs hỗ trợ nhiều (completion handlers, nhiều delegate, callbacks,…) | Chạy trên main thread |
| Animation asynchronous -> không lag |  | Animation cho nhiều loại object (View, Layer, Speaker volume,..) | UI có thể bị lag trong quá trình animate |
| Do Apple phát triển | | Nhiều animation động được xây dựng | Sử dụng nhiều CPU để tính toán |

* 1. Các thành phần của POP

