

表示周りのちょっとした事
cqa02303

1.表示変更とMainLoopの関係

- ・UIKitのプロパティ変更（描画パラメータ変更）はMainThreadで行う
→@property (nonatomic)だから（競合の解決よりも処理速度優先）
- ・変更した内容が更新されるのは今のメソッドを終了しMainLoopに戻った後
→直後に実行される保証は無い（キューの順位競合がある）
- ・重い処理を行いたい場合、表示変更+重い処理は遅延実行
→メインスレッドに切替えてperformSelector:withObject:afterDelay:
→ViewControllerのインスタンス生成すらも重い処理になる
→@selector()を使う場合は@interface節での宣言不要

2.Property Animation

- ・ QuartzCore Framework / #import "QuartzCore/QuartzCore.h"
- ・ beginAnimations:context: ~ commitAnimations の間がQueueになる
→contextはC pointerなのでキャストを理解すれば何でもあり
→animationIDはdelegate selectorでどのAnimationか識別するがselector変えたほうが早くネ？
- ・ StartDateやDelayは使いどころで役に立つ。
→残像的なモノとか、まとめて指定できるから
- ・ RepeatCountの繰り返し回数は整数では無い！
→Autoreversesを付けた時には初期状態まで戻った直後に指定状態になる為、0.5回追加する事で解消できる
（例：1→2のアニメーションの場合 / 1-(なめらか)->2-(なめらか)->1-(一瞬で切替)->2)
- ・ プロパティはアニメーション開始直後に終了値に変更される
→途中で現在の値を取得出来ない
→アニメーション処理中に同じプロパティのアニメーションを開始すると動作が断絶する
（例：1→2の途中で→3を指定 / 1--->1.n-(一瞬で切替)->2-(なめらか)->3)
→ただし、同じ終了値のアニメーションを指定した場合には「実行されない」
（この特性は反応の良いレスポンスを実現する為に活用できる）
- ・ 種類の異なるプロパティには別のアニメーションが共存できる
→例：こんなことも出来ます
 - ・ alpha：3秒でフェードイン
 - ・ center：4秒で画面外からスライド
 - ・ transform：2秒で0.9倍にした後stop selectorで1秒で1.0倍にバウンス
- ・ centerとtransformでの位置ずらしを組み合わせる事も出来る
- ・ OtherThreadからのbeginAnimationは処理されない事がある
→別のアニメーション実行時のdraw処理まで再描画されないような動き

3.UICollectionView Animation

- ・ UIKit版GIFアニメ
→複数の画像を指定してパラパラアニメ
→切替え間隔固定
→繰り返し回数指定可能
→ガラケー的ボタン出現処理とか？
- ・ animationImagesに画像配列を入れてstartAnimatingするだけ

4.AffineTransform

- ・初期状態に戻す場合、初期状態をMakeではなく定義された初期値を使う
→CGAffineTransformMake(0.0f,,,)ではなくCGAffineTransformIdentity
- ・90度回転はMakeRotationではなくMake(1,0,0,1)等を使う
→誤差が出やすいから (degree/radian変換、三角関数)
→反転も作りやすい
→誤差はそのまま中間点の混色処理になってしまう
- ・UIViewのtransformはアクセサがCALayerのtransformに変換して渡している (はず)
→UIViewは2次元平面用、CALayerは3D空間用
→CALayerではカバーフロー的台形変換が出来る

5.表示順序

- ・InterfaceBuilder
→リストの上が奥、貼り付けた順番に手前が一番下
- ・UIView
→[親view bringSubviewToFront:子view];
- ・CALayer
→CGFloat zPosition : 手前が正

6.表示の制限

- ・clipToBounds
→長方形の範囲内のみ表示
- ・Calayer cornerRadius
→角丸をソースのpx単位で指定 (resize時に注意)
→clipToBounds前提
- ・CALayer mask
→自由形状の範囲内のみ表示

5.アプリケーションからのリアクション

- ・表示を変更するタイミング
→操作からの呼出し
→タイマからの呼出し
→通信からの呼出し
- ・定期的に表示変更したい
→NSTimer
→Accereromator
→Animation stop selectorからチェーン
- ・平行するアニメーション
 - ・同タイミングのアニメーションがある場合、キューを集約する
(beginAnimation~commitAnimationの中で複数のviewを同時に操作する)
→beginAnimationは別々に管理されたタイマみたいな物と考えて良い様です
- ・あまりに重いと画面再描画が間に合わなくなる