iPhone Programming
Animations

Tim Gegg-Harrison

Animation

- The geometric properties of a view can be animated fairly easily
- `UIView` provides several class methods that can be used to perform simple animations such as moving a view instance to a new position or enlarging it.

Animation …

- The way to run animations is using the following `UIView` methods

```
+ animate:withDuration:delay:options:animations:completion:
+ animate:withDuration:animations:completion:
+ animate:withDuration:animations:
```

- Duration:
  - The total duration of the animations, measured in seconds. If you specify a negative value or 0, the changes are made without animating them.

- Delay:
  - The amount of time (measured in seconds) to wait before beginning the animations. Specify a value of 0 to begin the animations immediately.

Animation …

- The way to run animations is using the following `UIView` methods

```
+ animate:withDuration:delay:options:animations:completion:
+ animate:withDuration:animations:completion:
+ animate:withDuration:animations:
```

- Options:
  - A mask of options indicating how you want to perform the animations.

```
UIViewAnimationOptionTransition.none           = 0 << 20,
UIViewAnimationOptionTransition.flipFromLeft   = 1 << 20,
UIViewAnimationOptionTransition.flipFromRight  = 2 << 20,
UIViewAnimationOptionTransition.curlUp         = 3 << 20,
UIViewAnimationOptionTransition.curlDown       = 4 << 20,
UIViewAnimationOptionTransition.crossDissolve  = 5 << 20,
UIViewAnimationOptionTransition.flipFromTop    = 6 << 20,
UIViewAnimationOptionTransition.flipFromBottom = 7 << 20
```

Animation …

- The way to run animations is using the following `UIView` methods

```
+ animate:withDuration:delay:options:animations:completion:
+ animate:withDuration:animations:completion:
+ animate:withDuration:animations:
```

- Animations:
  - A closure object containing the changes to commit to the views. This is where you programmatically change any animatable properties of the views in your view hierarchy. This closure takes no parameters and has no return value. This parameter must not be nil.

Animation …

- The way to run animations is using the following `UIView` methods

```
+ animate:withDuration:delay:options:animations:completion:
+ animate:withDuration:animations:completion:
+ animate:withDuration:animations:
```

- Completion:
  - A closure object to be executed when the animation sequence ends. This block has no return value and takes a single Boolean argument that indicates whether or not the animations actually finished before the completion handler was called. If the duration of the animation is 0, this closure is performed at the beginning of the next run loop cycle. This parameter may be nil.

Animation …

```
let bottomPoint: CGPoint = CGPoint(x: ball.center.x,
  y: (UIScreen.main.bounds.size.height-(self.ball.bounds.width/2)))
let topPoint: CGPoint = CGPoint(x: basketball.center.x,
  y: 0.75*ball.center.y+0.25*bottomPoint.y)
UIView.animate(withDuration: 4,
    animations: {
       () -> Void in
           self.ball.center = bottomPoint
    },
    completion: {
       (Bool) -> Void in
           UIView.animate(withDuration: 3,
             animations: {
                () -> Void in
                   self.ball.center = topPoint
             },
             completion: {
                (Bool) -> Void in

             })
    })
```