Core Animation

DevDay for iPhone
Daniel H Steinberg
dimsumthinking

Use Animation

- To set context for the user
- To help the user make transitions
- To build a mental and physical model of the space in which they use your app
- To fit in with user expectation of apps that run on the platform

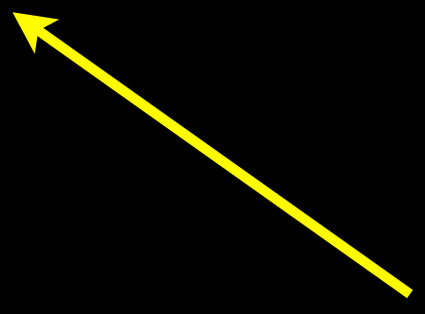
Canonical Example

- Check out the various animations enabled in a standard Apple table view example.

Implementing the animation

```
-   (void)    tableView:(UITableView *)tableView
didSelectRowAtIndexPath:(NSIndexPath *)indexPath {

  DetailViewController *detailViewController =
    [[DetailViewController alloc]
             initWithNibName:@"DetailViewController" bundle:nil];
  detailViewController.timeStamp =
    [[[self.fetchedResultsController objectAtIndexPath:indexPath]
                                          valueForKey:@"timeStamp"]
                                               description];
  [self.navigationController pushViewController:detailViewController
                                       animated:YES];
  [detailViewController release];
}
```



Basic Tables/RootViewController.m

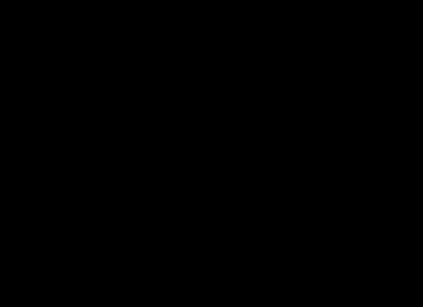
Should we leave it out?

Basic TablesNoAnimation

Not Implementing the animation

```
-   (void)    tableView:(UITableView *)tableView
didSelectRowAtIndexPath:(NSIndexPath *)indexPath {

  DetailViewController *detailViewController =
    [[DetailViewController alloc]
             initWithNibName:@"DetailViewController" bundle:nil];
  detailViewController.timeStamp =
    [[[self.fetchedResultsController objectAtIndexPath:indexPath]
                                          valueForKey:@"timeStamp"]
                                           description];
  [self.navigationController pushViewController:detailViewController
                                       animated:NO];
  [detailViewController release];
}
```



Basic TablesNoAnimation/RootViewController.m

Animation is important

but ...

it is not today's blink tag

Basic TablesFlipAnimation

On Using Animation

- Why
- When
- How

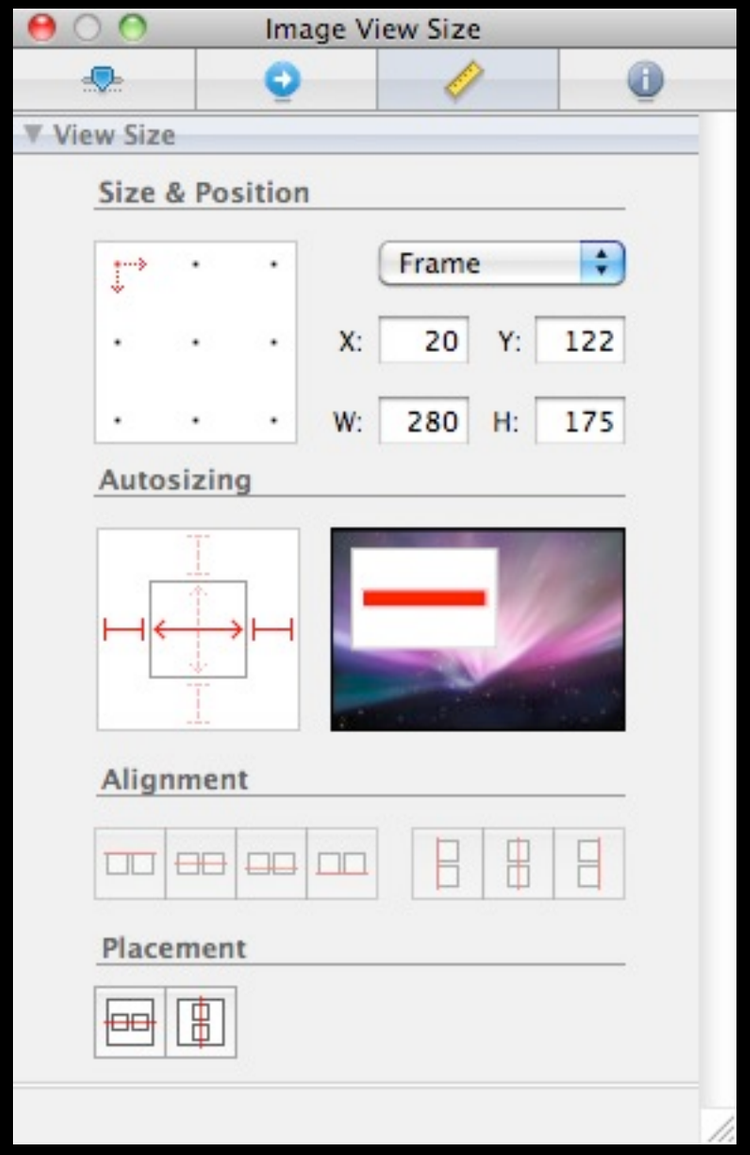
Don’t fight the platform

```
animated:YES
```

Split-View Template

Use the size inspector

```
- (BOOL)shouldAutorotateToInterfaceOrientation:
                (UIInterfaceOrientation)interfaceOrientation {
    return YES;
}
```



Simple Rotation

Let's do this in code

```
-(void)drawPortraitVersionOfScreen {
    self.imageView.frame = CGRectMake(20, 120, 280, 175);
    self.view.backgroundColor = [UIColor blackColor];
}
-(void)drawLandscapeVersionOfScreen {
    self.imageView.frame = CGRectMake(60, 45, 350, 220);
    self.view.backgroundColor = [UIColor whiteColor];
}
-(void)didAnimateFromInterfaceOrientation:
                                (UIInterfaceOrientation)interfaceOrientation
                                   duration:(NSTimeInterval)duration {
    if (interfaceOrientation == UIInterfaceOrientationPortrait ||
        interfaceOrientation == UIInterfaceOrientationPortraitUpsideDown) {
        [self drawLandscapeVersionOfScreen];
    } else {
        [self drawPortraitVersionOfScreen];
    }
}
```

Non Rotation In Code

Let's really^do this in code

```
-(void)drawPortraitVersionOfScreen {
    self.imageView.frame = CGRectMake(20, 120, 280, 175);
    self.view.backgroundColor = [UIColor blackColor];
}
-(void)drawLandscapeVersionOfScreen {
    self.imageView.frame = CGRectMake(60, 45, 350, 220);
    self.view.backgroundColor = [UIColor whiteColor];
}
- (void)willAnimateRotationToInterfaceOrientation:
(UIInterfaceOrientation)interfaceOrientation
duration:(NSTimeInterval)duration {
    if (interfaceOrientation == UIInterfaceOrientationPortrait ||
        interfaceOrientation == UIInterfaceOrientationPortraitUpsideDown) {
        [self drawPortraitVersionOfScreen];
    } else {
        [self drawLandscapeVersionOfScreen];
    }
}
```

Rotation in Code

Initiating the change yourself *

```
- (void)viewDidLoad {
    [super viewDidLoad];
    [self displayReadyImageOnScreen];
    [self.view addGestureRecognizer: [[[UITapGestureRecognizer alloc]
                                initWithTarget:self
                                        action:@selector(viewDidGetTapped)]
                                        autorelease]];
}
- (void)viewDidGetTapped {
    if (isDisplayingAtBottom) {
        [self displayAtTop];
    } else {
        [self displayAtBottom];
    }
    isDisplayingAtBottom = !isDisplayingAtBottom;
}
```

Gesture Non Animated

* in iPhone 3.2 and iOS 4

Animating the change

```
-(void)displayAtTop {
    [UIView beginAnimations:@"move to top" context:nil];
    self.imageView.frame = CGRectMake(20, 0, 280, 175);
    self.view.backgroundColor = [UIColor blackColor];
    [UIView commitAnimations];
}
```

Gesture Animated

Slowing things down

```
-(void)displayAtTop {
    [UIView beginAnimations:@"move to top" context:nil];
    [UIView setAnimationDuration:4];
    self.imageView.frame = CGRectMake(20, 0, 280, 175);
    self.view.backgroundColor = [UIColor blackColor];
    [UIView commitAnimations];
}
```

Gesture Slow Animated

Changing the profile

```
-(void)displayAtTop {
    [UIView beginAnimations:@"move to top" context:nil];
    [UIView setAnimationDuration:4];
    [UIView setAnimationCurve:UIViewAnimationCurveEaseIn];
    self.imageView.frame = CGRectMake(20, 0, 280, 175);
    self.view.backgroundColor = [UIColor blackColor];
    [UIView commitAnimations];
}
```

Gesture Ease In Animated

Using blocks *

```
-(void)displayAtTop {
    [UIView animateWithDuration:4 animations:^{
        self.imageView.frame = CGRectMake(20, 0, 280, 175);
        self.view.backgroundColor = [UIColor blackColor];
    }];
}
-(void)displayAtBottom {
    [UIView beginAnimations:@"move to bottom" context:nil];
    [UIView setAnimationDuration:4];
    [UIView setAnimationCurve:UIViewAnimationCurveEaseIn];
    self.imageView.frame = CGRectMake(60, 300, 210, 130);
    self.view.backgroundColor = [UIColor whiteColor];
    [UIView commitAnimations];
}
```

Simple Block Animation

* in iOS 4

Adding Options

```
-(void)displayAtTop {
    [UIView animateWithDuration:4
                          delay:0
                        options:UIViewAnimationCurveEaseIn
                     animations:^{
        self.imageView.frame = CGRectMake(20, 0, 280, 175);
        self.view.backgroundColor = [UIColor blackColor];
    }
                     completion:nil];
}
```

Medium Block Animation

More Options

```
-(void)displayAtTop {
    [UIView animateWithDuration:4
                          delay:0
                        options:UIViewAnimationCurveEaseIn +
     UIViewAnimationOptionBeginFromCurrentState +
     UIViewAnimationOptionAllowUserInteraction
                     animations:^{
                         self.imageView.frame = CGRectMake(20, 0, 280, 175);
                         self.view.backgroundColor = [UIColor blackColor];
                     }
                     completion:nil];
}
```

Medium Block Animation Plus

Adding a completion

```
-(void)displayAtTop {
    [UIView animateWithDuration:4
                          delay:0
                        options:UIViewAnimationCurveEaseIn
                     animations:^{
                         self.imageView.frame = CGRectMake(20, 0, 280, 175);
                         self.view.backgroundColor = [UIColor blackColor];
                         self.imageView.alpha = .4;
                     }
                     completion:^(BOOL finished){
                         [UIView animateWithDuration:3 animations:^{
                             self.imageView.alpha = 1.0;
                         }];
                     }];
}
```

Completion Block Animation

Refactor

```
-(void) animate:(void (^)(void))animations
 withCompletion:(void (^)(BOOL finished))completion {
    [UIView animateWithDuration:4
                          delay:0
                        options:UIViewAnimationCurveEaseIn +
     UIViewAnimationOptionBeginFromCurrentState +
     UIViewAnimationOptionAllowUserInteraction
                     animations:animations
                     completion:completion];
}
```

```
-(void)displayAtTop {
    [self animate:^{
        self.imageView.frame = CGRectMake(20, 0, 280, 175);
        self.view.backgroundColor = [UIColor blackColor];
        self.imageView.alpha = .4;
    }
   withCompletion:^(BOOL finished){
        [UIView animateWithDuration:3 animations:^{
            self.imageView.alpha = 1.0;
        }];
   }];
}
```

Completion Block Animation Refactor

Simpler Refactor

```
-(void) animate:(void (^)(void))animations {
    [UIView animateWithDuration:4
                          delay:0
                        options:UIViewAnimationCurveEaseIn +
     UIViewAnimationOptionBeginFromCurrentState +
     UIViewAnimationOptionAllowUserInteraction
                     animations:animations
                     completion:nil];
}
```

```
-(void)displayAtTop {
    [self animate:^{
        self.imageView.frame = CGRectMake(20, 0, 280, 175);
        self.view.backgroundColor = [UIColor blackColor];
    }];
}

-(void)displayAtBottom {
    [self animate:^{
        self.imageView.frame = CGRectMake(60, 300, 210, 130);
        self.view.backgroundColor = [UIColor whiteColor];
    }];
}
```

Completion Block Animation Refactor2

Changing images

```
-(void)displayAtTop {
    [UIView animateWithDuration:4 animations:^{
        self.imageView.frame = CGRectMake(20, 0, 280, 175);
        self.imageView.image = [UIImage imageNamed:@"DevDayReady.png"];
    }];
}
-(void)displayAtBottom {
    [UIView animateWithDuration:4 animations:^{
        self.imageView.frame = CGRectMake(0, 410, 320, 50);
        self.imageView.image = [UIImage imageNamed:@"DevDayBanner.png"];
    }];
}
```

Simple Non Layer Animation

Working with Layers

```
-(void)displayAtTop {
    self.imageLayer.frame = CGRectMake(20, 0, 280, 175);
    self.imageLayer.contents = (id)([UIImage
imageNamed:@"DevDayReady.png"].CGImage);

}
-(void)displayAtBottom {
    self.imageLayer.frame = CGRectMake(0, 410, 320, 50);
    self.imageLayer.contents = (id)( [UIImage
imageNamed:@"DevDayBanner.png"].CGImage);
}
- (void)viewDidLoad {
    [super viewDidLoad];
    [self.view addGestureRecognizer:
     [[[UITapGestureRecognizer alloc]
       initWithTarget:self
       action:@selector(viewDidGetTapped)]
      autorelease]];
    self.imageLayer = [CALayer layer];
    [self.view.layer addSublayer:self.imageLayer];
    [self displayAtTop];
}
```

Simple Layer Animation

Freeze

- Start simple
- Keep users in mind
- Get complex when you need to but...
- Never when you don’t need to

And beyond ...

- Code and slides:
- http://dimsumthinking.com/services/code/