Advanced Scroll View Techniques

Session 104

Josh Shaffer and Eliza Block
iOS Frameworks and Applications

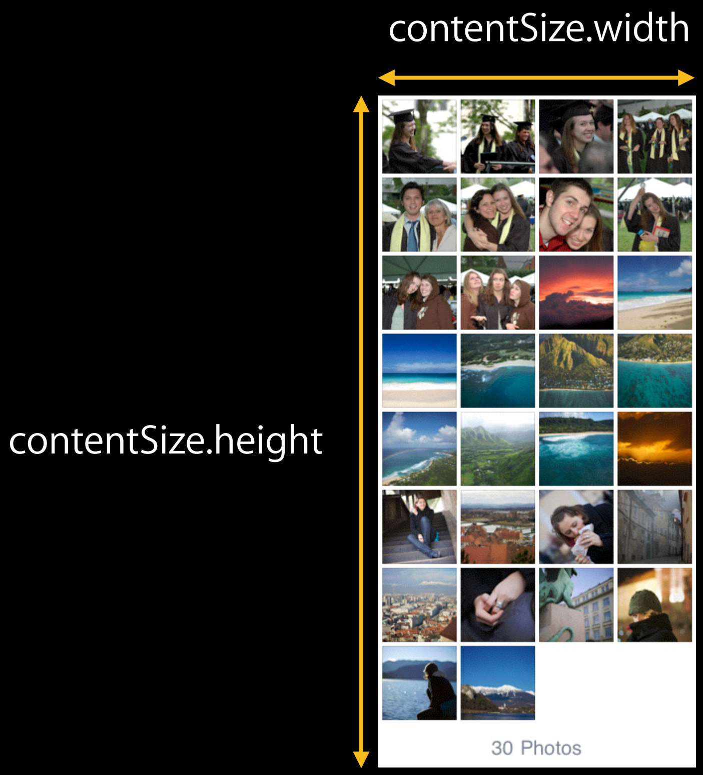
Advanced Scroll View Techniques

- Infinite scrolling
- Stationary views
- Custom touch handling
- Redraw after zooming

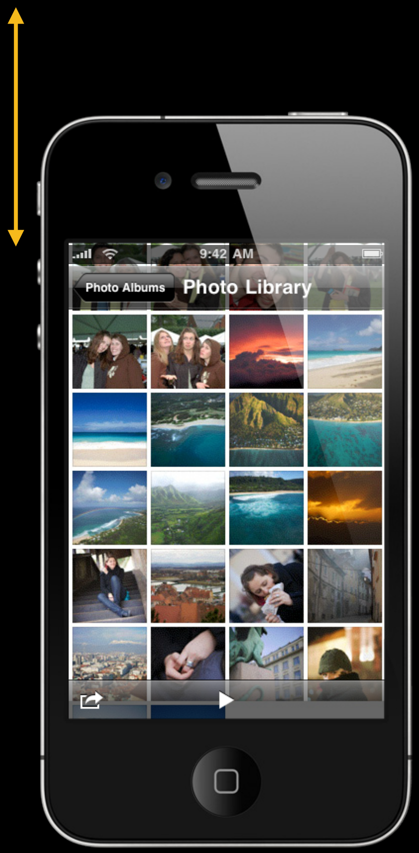
Configure

Review of the basics

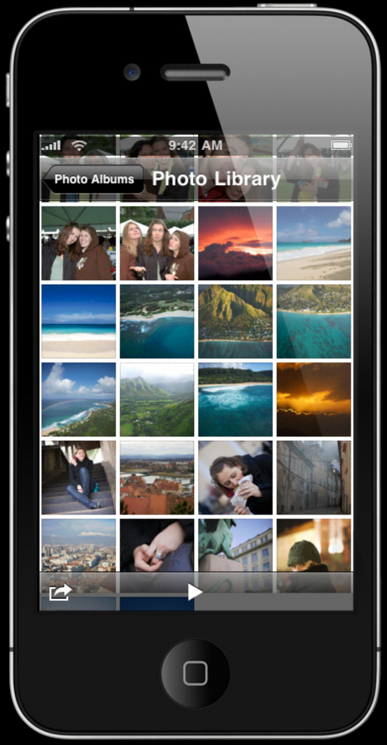
Content Size



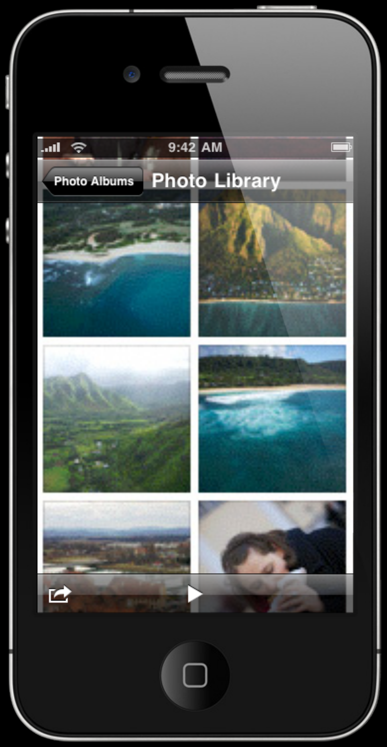
Content Offset



View for Zooming in Scroll View

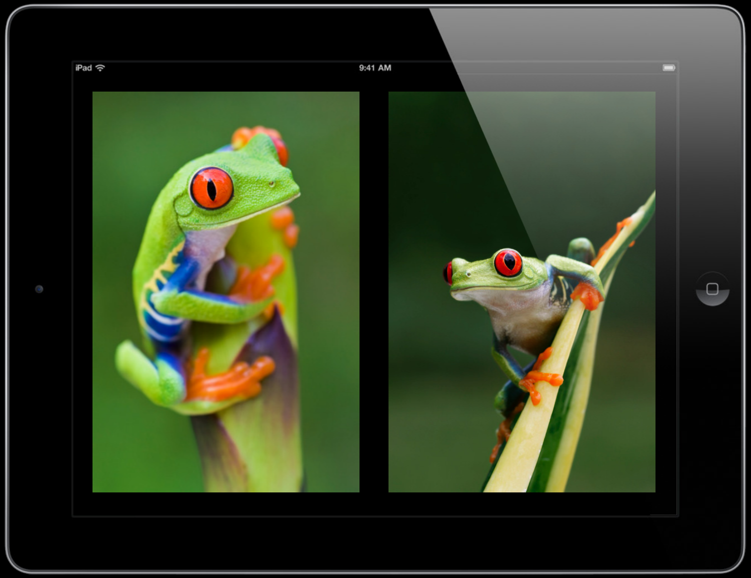


View for Zooming in Scroll View

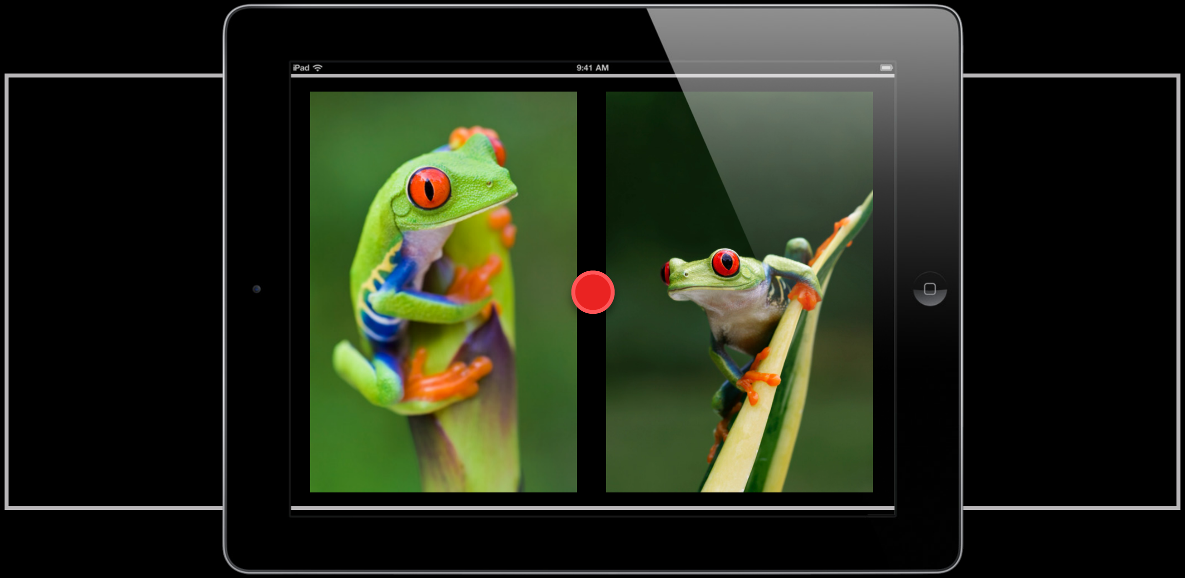


Infinite Scrolling

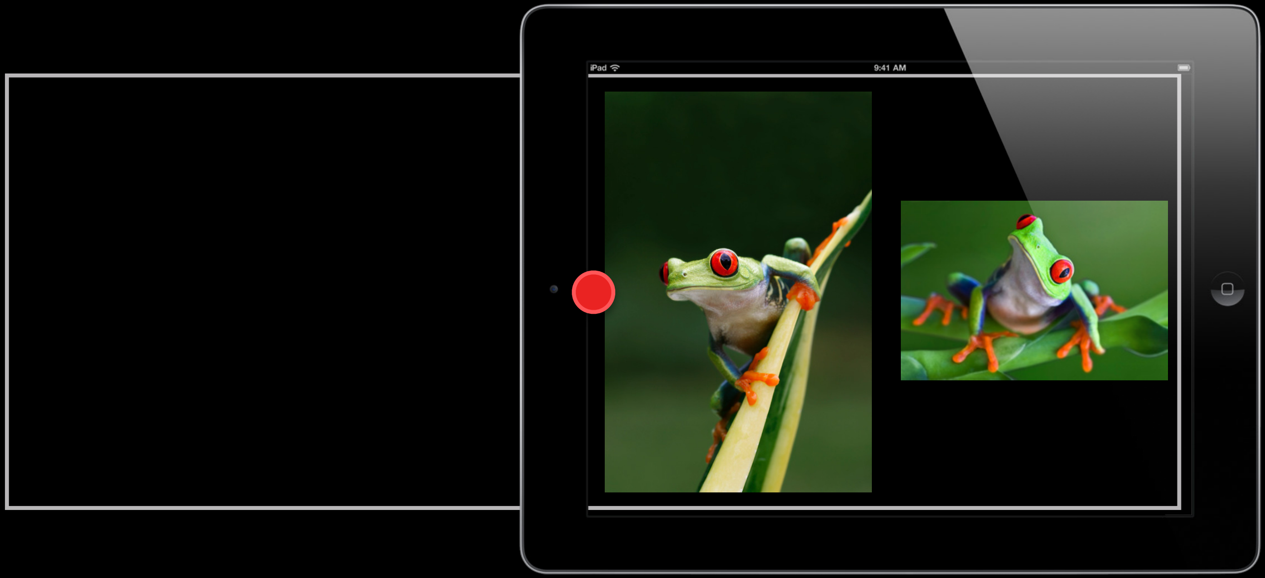
Infinite Scrolling



Infinite Scrolling

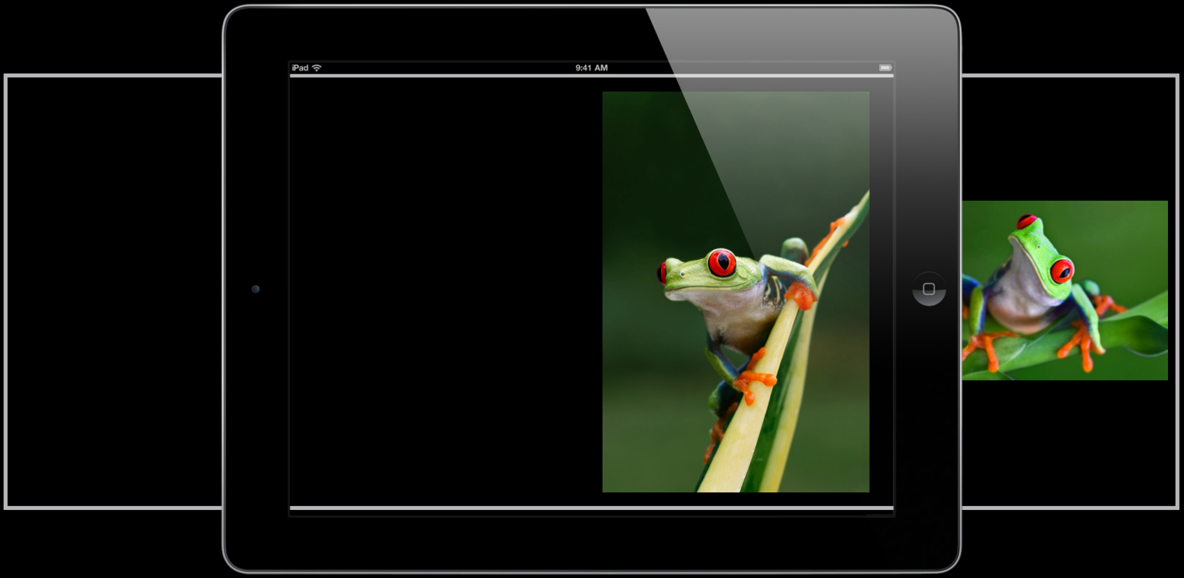


Infinite Scrolling



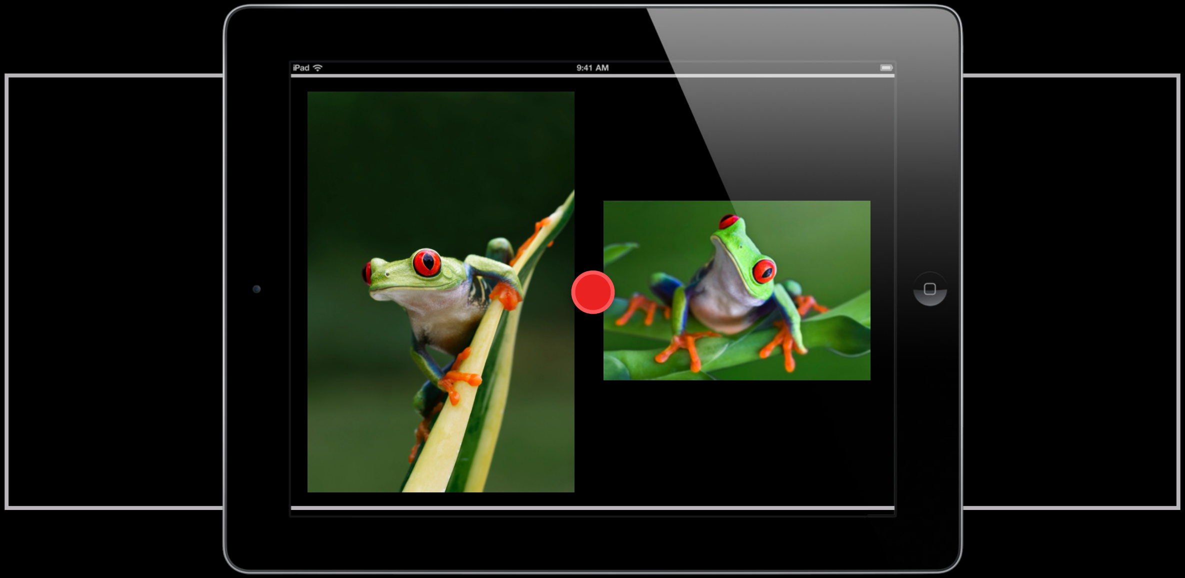
Infinite Scrolling

Adjust contentOffset

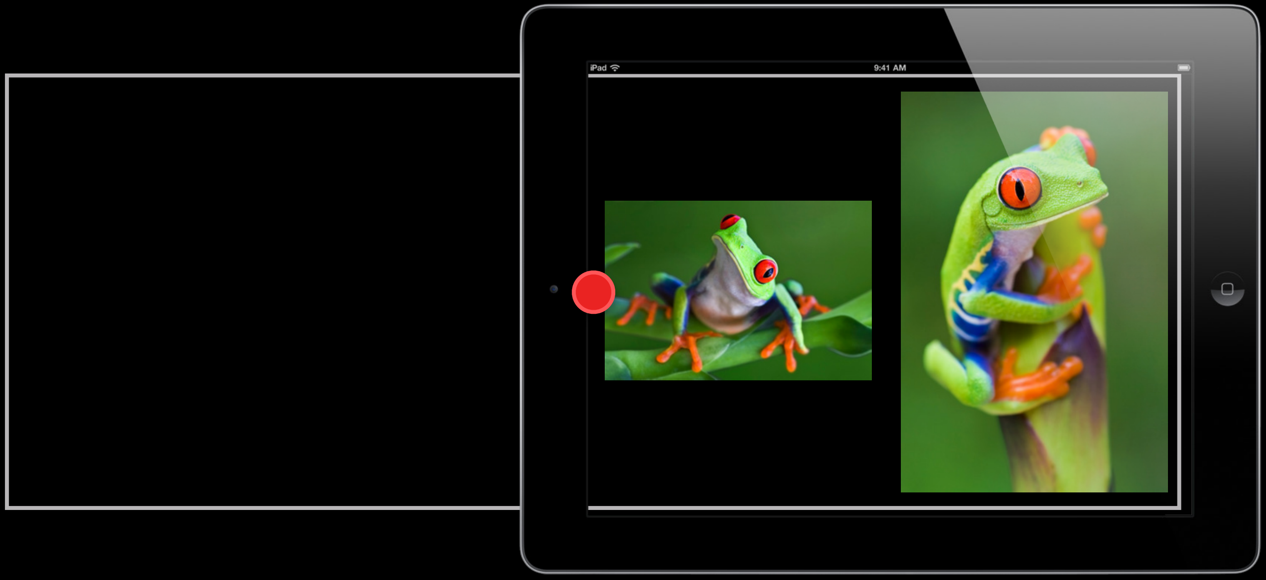


Infinite Scrolling

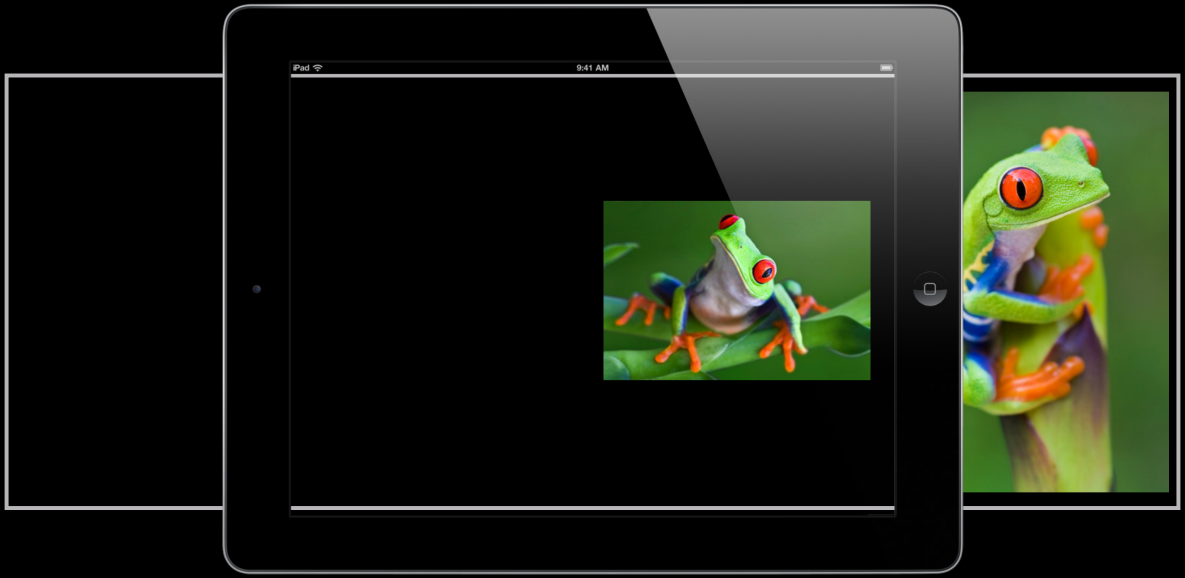
Adjust subview frames



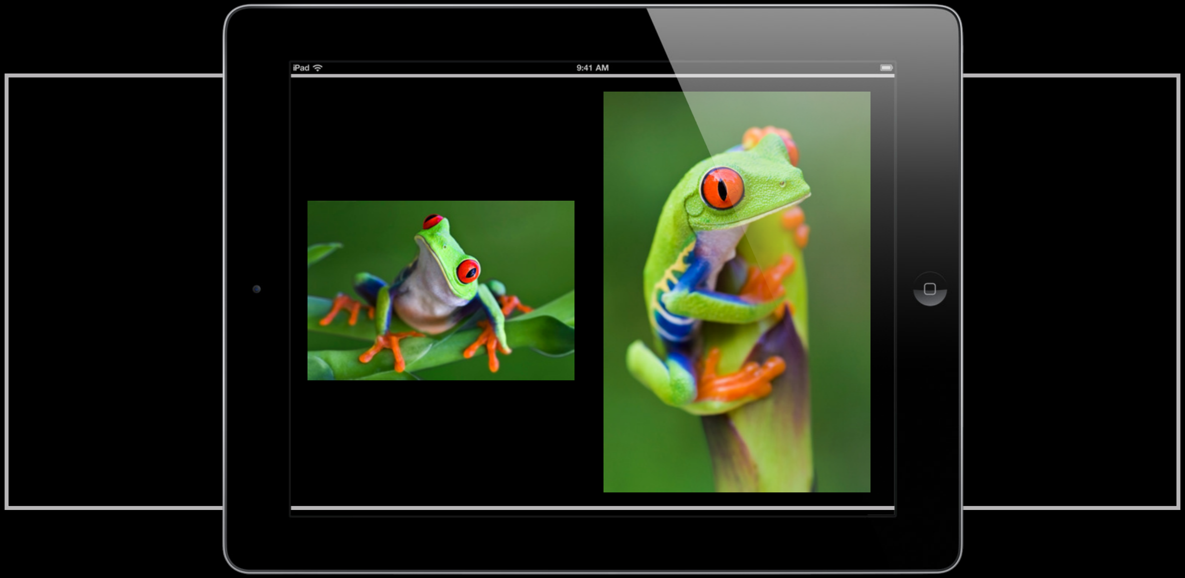
Infinite Scrolling



Infinite Scrolling



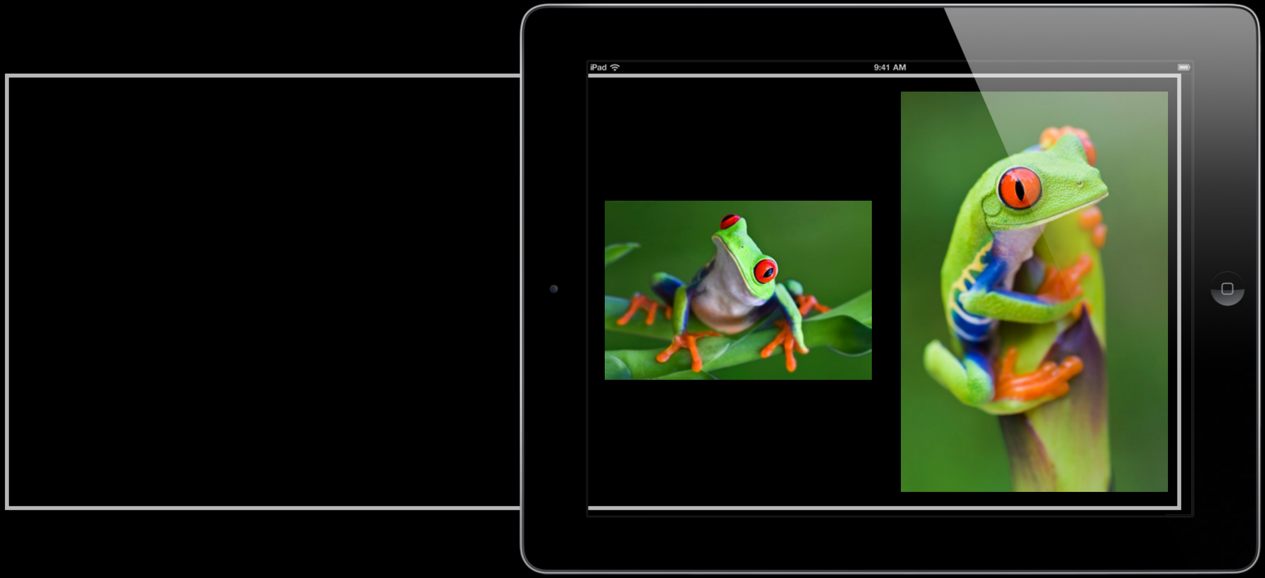
Infinite Scrolling



Infinite Scrolling

Where it happens

- Subclass and override layoutSubviews
- Recenter with setContentOffset:
- Shift subviews by the same amount with setFrame: or setCenter:



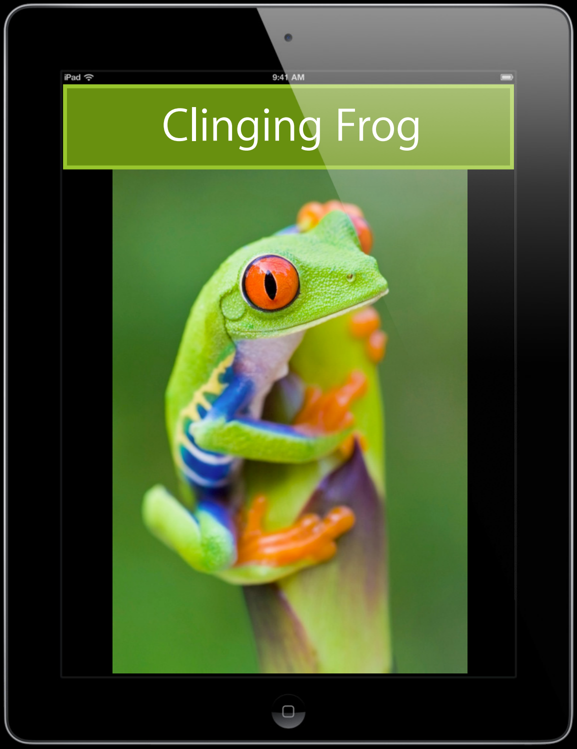
Demo

Eliza Block

Stationary Views and Header Views

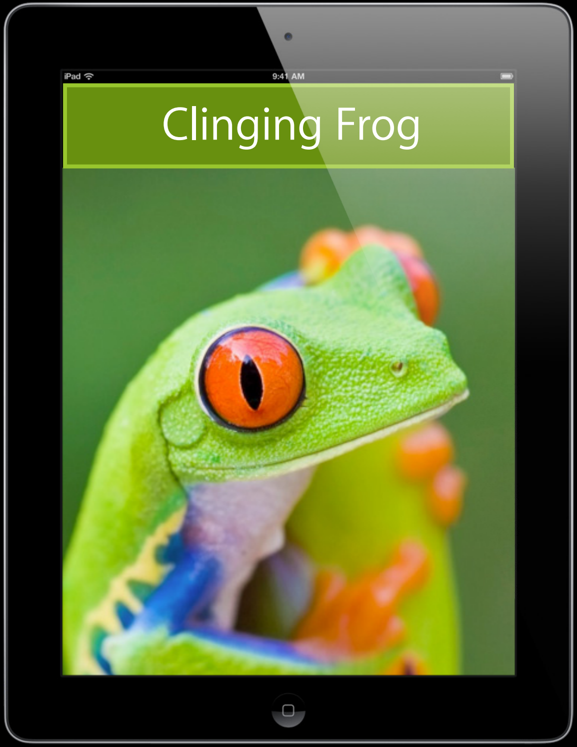
Another reason to love layoutSubviews

Header Views



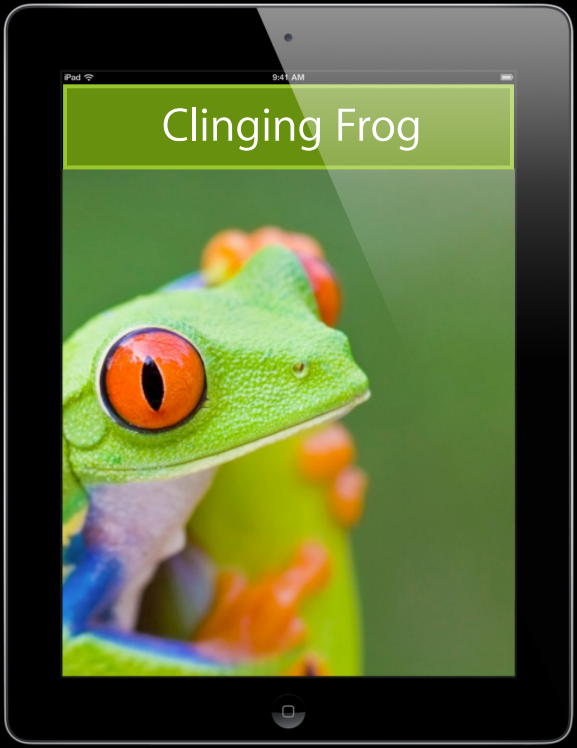
Header Views

Avoid zooming



Header Views

Avoid horizontal scrolling

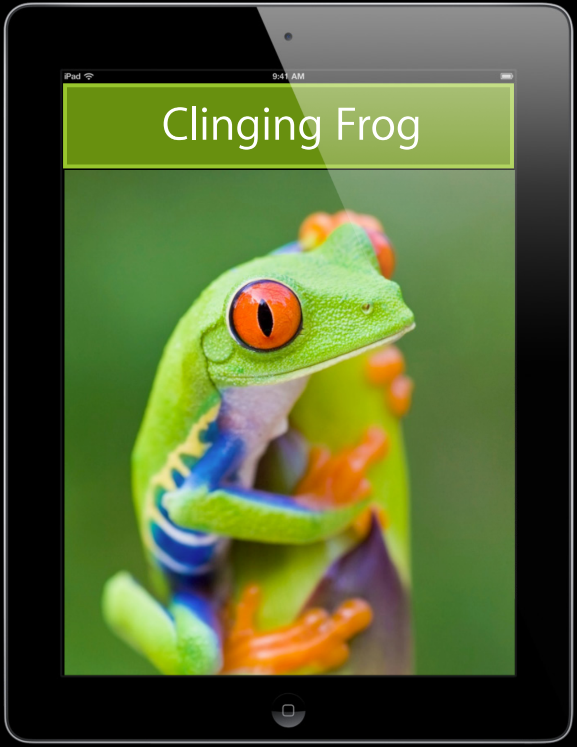


Header Views

Allow vertical scrolling

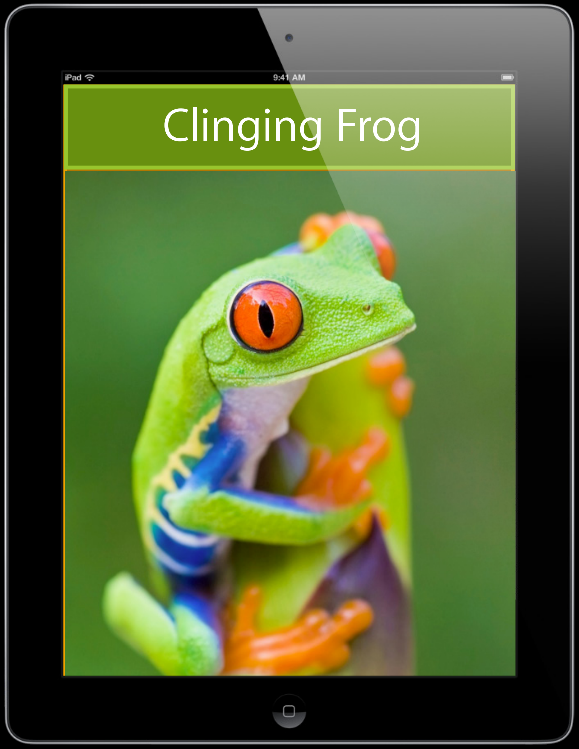
Header Views

Avoid zooming



Header Views

View configuration

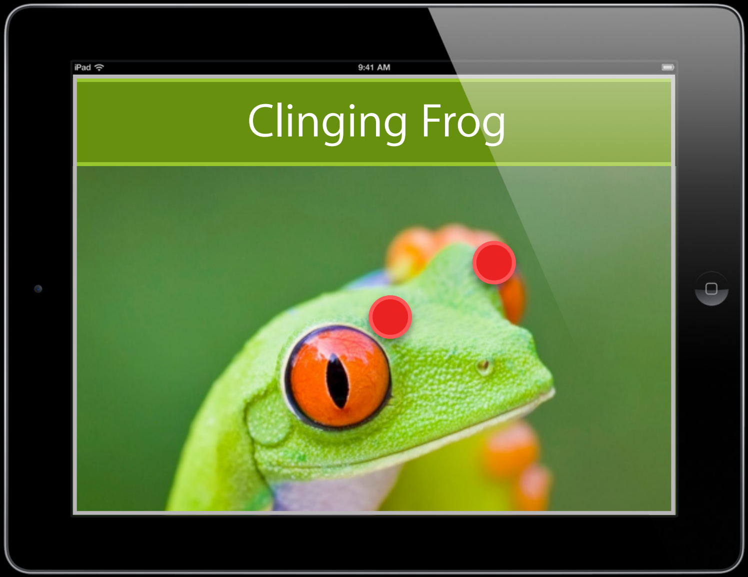


Header view
Sibling of viewForZoomingInScrollView

UIImageView
Returned from viewForZoomingInScrollView

Header Views

Pinning a view in place



Header Views

Pinning a view in place

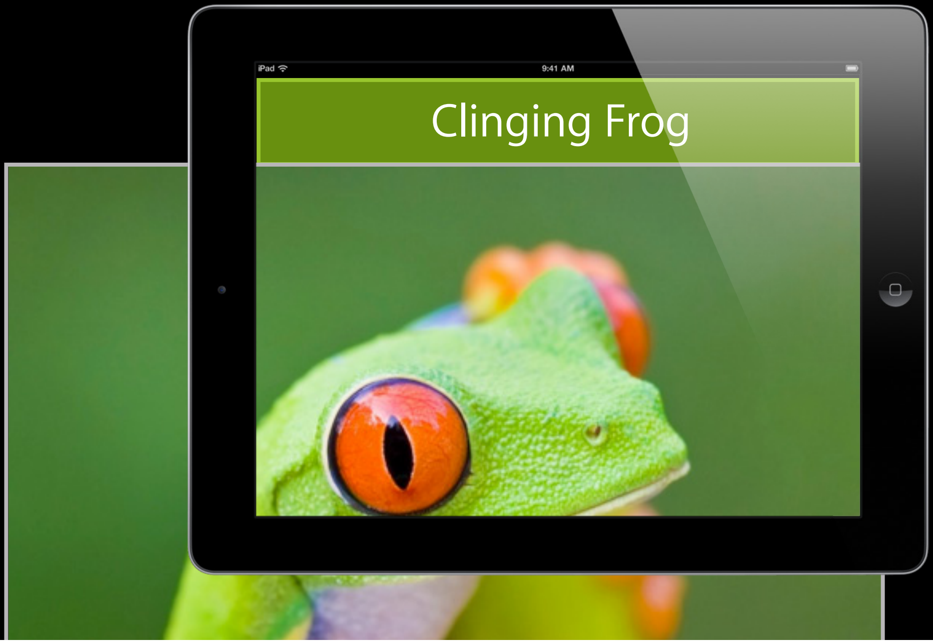
- Set header’s frame.origin.x to scroll view’s contentOffset.x

ging Frog
9:41 AM
iPad

Header Views

Pinning a view in place

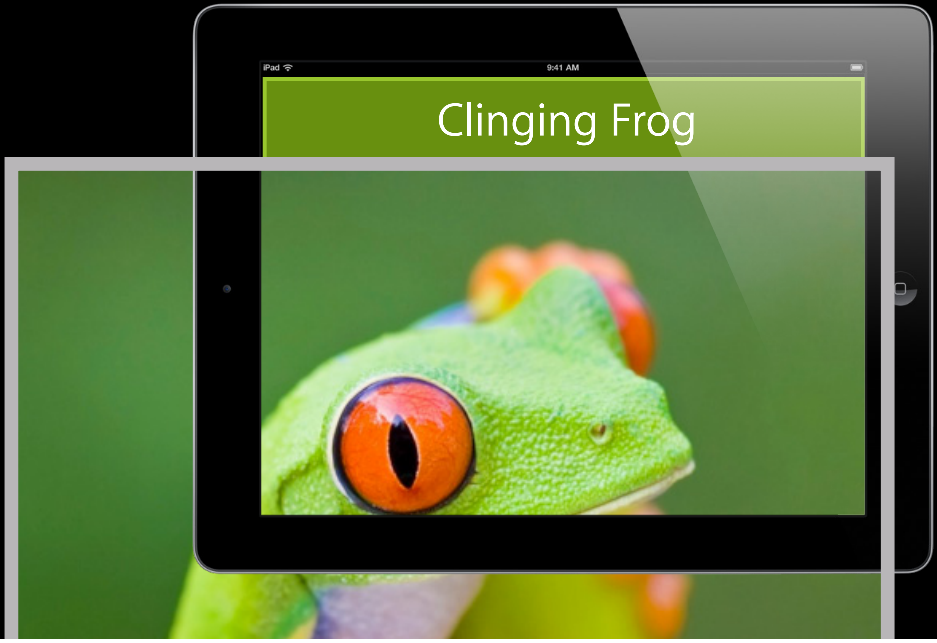
- Set header's frame.origin.x to scroll view's contentOffset.x



Header Views

Pinning a view in place

- Set header's frame.origin.x to scroll view's contentOffset.x
- Subclass UIScrollView to override setContentSize:
    - Calculate an appropriate, larger contentSize



Demo

Eliza Block

Customizing Touch Handling

Scroll views and gesture recognizers

Scrolling and Swiping

Failure requirements

Scrolling and Swiping

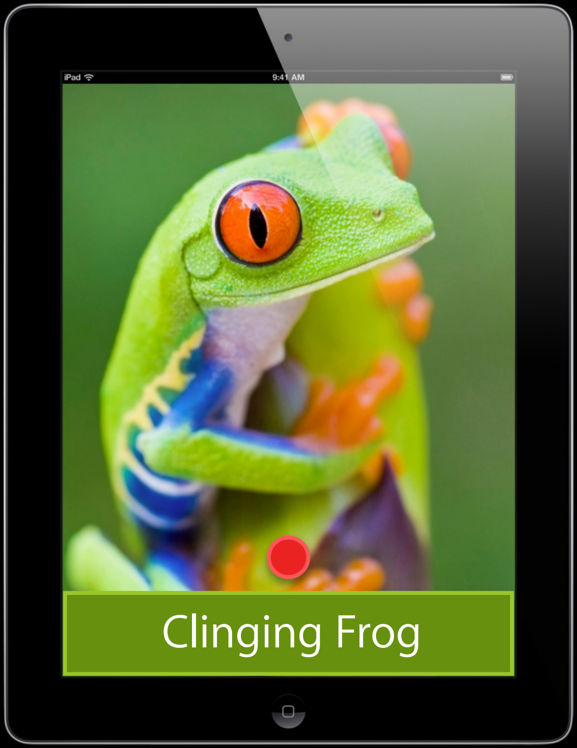
Failure requirements

Scrolling and Swiping

Failure requirements

Scrolling and Swiping

Failure requirements



UIScrollView and UIGestureRecognizer

Customizing user interaction

```
@property (readonly) UIPanGestureRecognizer   panGestureRecognizer;
@property (readonly) UIPinchGestureRecognizer pinchGestureRecognizer;
```

- Uses gesture recognizers for scrolling and zooming
- Allows interaction with other gesture recognizers

UIScrollView and UIGestureRecognizer

Swiping in a scroll view

```
UIScrollView *scrollView = [self scrollView];
UISwipeGestureRecognizer *swipeUp = [[UISwipeGestureRecognizer alloc]
                    initWithTarget:self action:@selector(handleSwipeUp:)];

swipeUp.direction = UISwipeGestureRecognizerDirectionUp;

[scrollView addGestureRecognizer:swipeUp];
[scrollView.panGestureRecognizer requireGestureRecognizerToFail:swipeUp];
```

UIScrollView and UIGestureRecognizer

Swiping in a scroll view

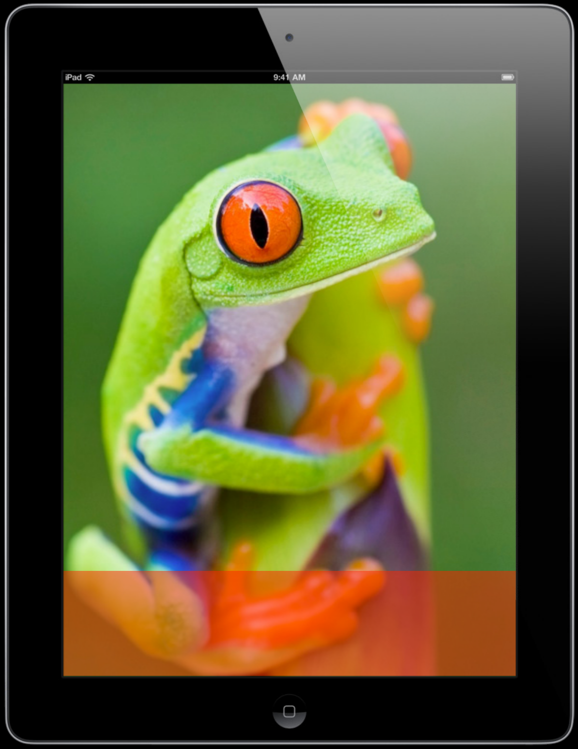
```
UIScrollView *scrollView = [self scrollView];
UISwipeGestureRecognizer *swipeUp = [[UISwipeGestureRecognizer alloc]
                    initWithTarget:self action:@selector(handleSwipeUp:)];

swipeUp.direction = UISwipeGestureRecognizerDirectionUp;

[scrollView addGestureRecognizer:swipeUp];
[scrollView.panGestureRecognizer requireGestureRecognizerToFail:swipeUp];
```

Scrolling and Swiping

Limiting target area

Scrolling and Swiping

Limiting target area



Scrolling and Swiping

Limiting target swipe area

```
- (BOOL)gestureRecognizer:(UIGestureRecognizer *)gestureRecognizer
       shouldReceiveTouch:(UITouch *)touch
{
    UIScrollView *scrollView = [self scrollView];
    CGRect  visibleBounds = [scrollView bounds];
    CGPoint touchPoint = [touch locationInView:scrollView];

    if (touchPoint.y < CGRectGetMaxY(visibleBounds) - 75)
        return NO;

    return YES;
}
```

Demo

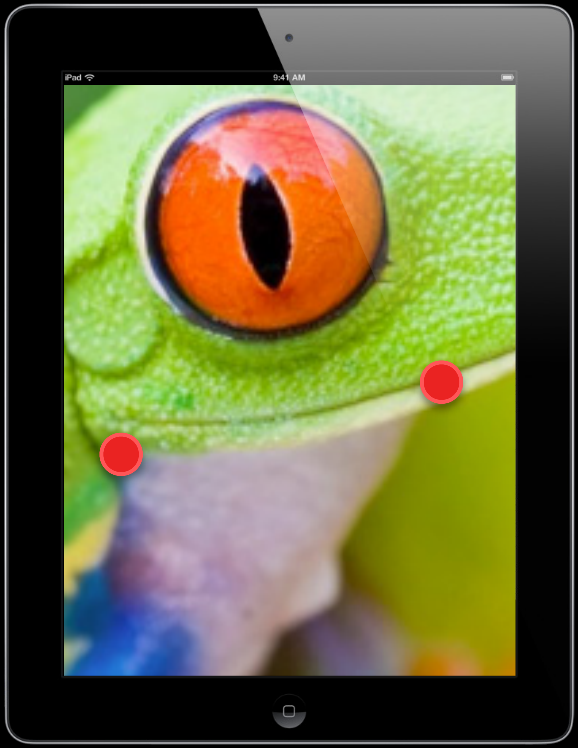
Eliza Block

Redraw After Zooming

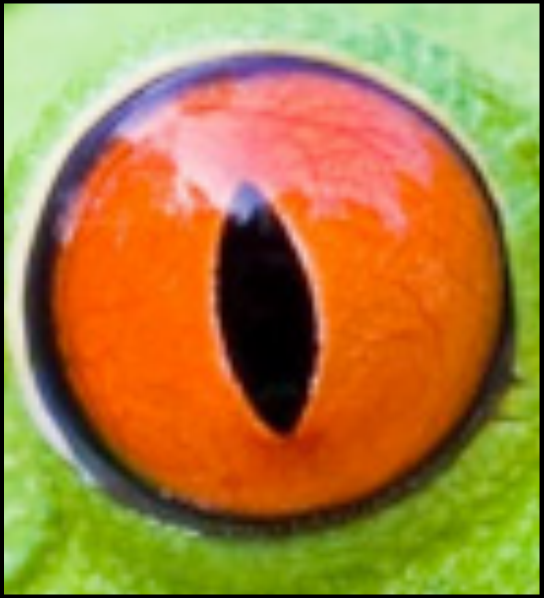
Getting crisp

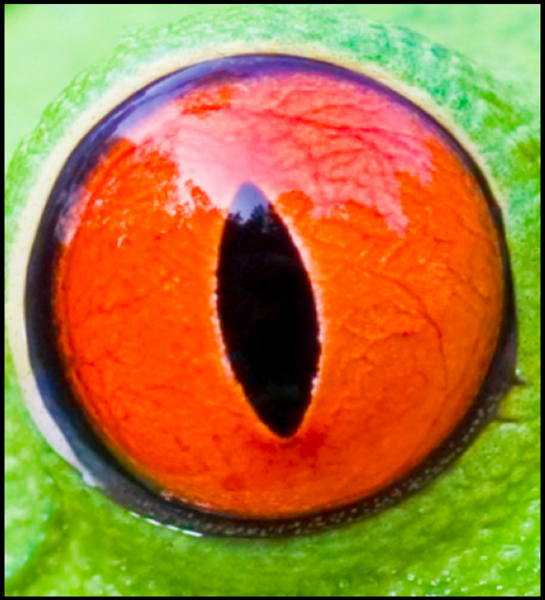
Redraw After Zooming

Redraw After Zooming



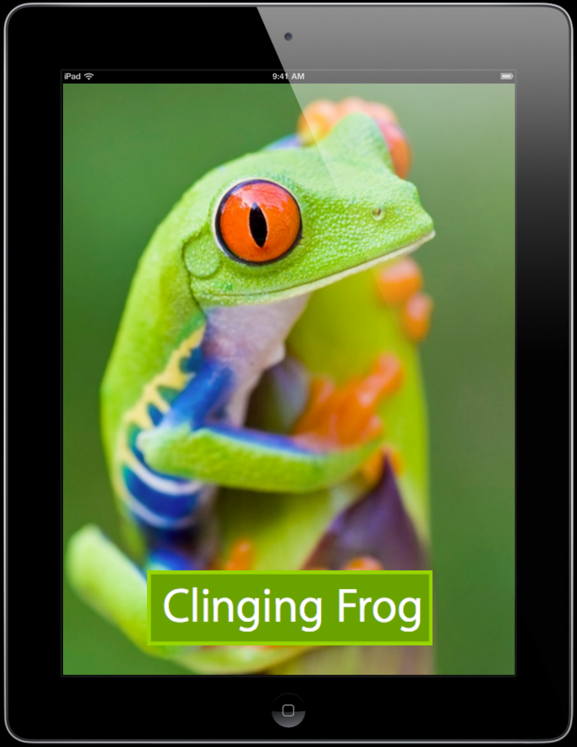
Redraw After Zooming





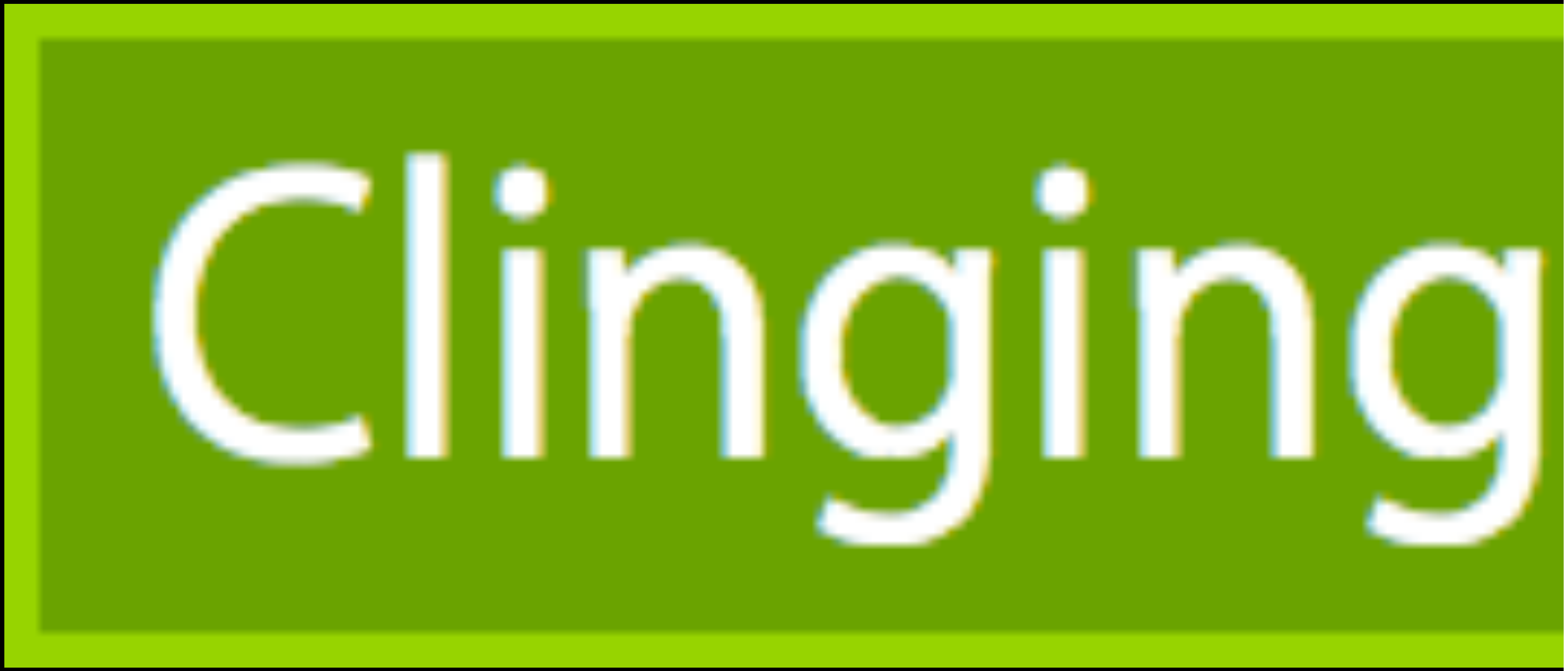
Redraw After Zooming

Easy trick for small content



Redraw After Zooming

Content scale factor



Redraw After Zooming

Content scale factor

Redraw After Zooming

```
-(void)scrollViewDidEndZooming:(UIScrollView *)sv
                      withView:(UIView *)view
                       atScale:(float)scale
{
    [view setContentScaleFactor:scale];
}
```

Redraw After Zooming

```
-(void)scrollViewDidEndZooming:(UIScrollView *)sv
                       withView:(UIView *)view
                        atScale:(float)scale
{
    scale *= [[[scrollView window] screen] scale];

    [view setContentScaleFactor:scale];
}
```

Demo

Eliza Block

More Information

Bill Dudney
Application Frameworks Evangelist
dudney@apple.com

Documentation
Scroll View Programming Guide for iOS
http://developer.apple.com/library/ios/#documentation/WindowsViews/Conceptual/UIScrollView_pg

Apple Developer Forums
http://devforums.apple.com

Related Sessions

UITableView Changes, Tips & Tricks	Nob HIll Thursday 2:00–3:00PM

Labs

UIScrollView Lab	Application Frameworks Lab B Tuesday 4:30PM
Cocoa Touch Lab	Application Frameworks Lab D Wednesday 2:00PM

WWDC2011