Building a vertical tab bar controller for the iPad

Nicolas Seriot
@nst021

May 19th, 2010
macprofessionals.ch user group, Bern

UISplitViewController

UITabController Usability Issues

useit.com Alertbox May 2010 iPad Usability

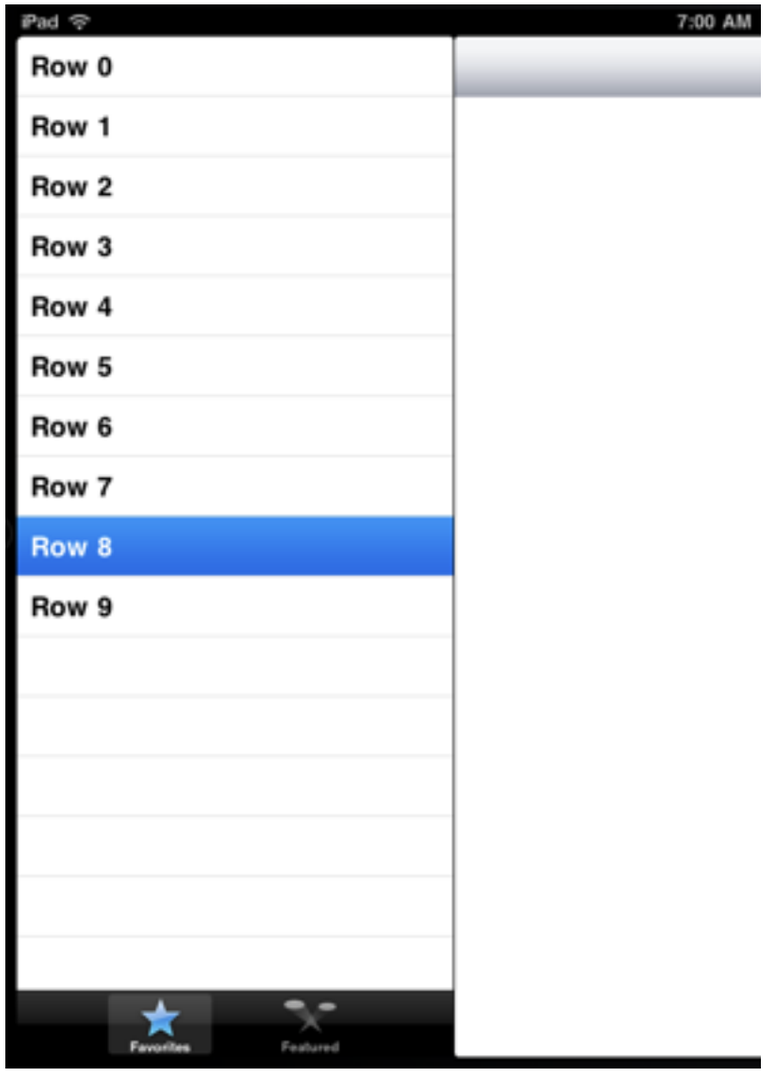
Jakob Nielsen's Alertbox, May 10, 2010:

iPad Usability: First Findings From User Testing

Indeed, one finding from our study is that **the tab bar at the bottom of the screen works much worse on iPad than on iPhone**. On the small phone, users are likely to notice the muted icons at the bottom of the screen, even if their attention is on content in the middle of the screen. But the iPad's much bigger screen means that users are typically directing their gaze far from the tab bar and they ignore (and forget) those buttons.

[...]

We know from cognitive psychology that **things farther away from the focus of attention tend to be ignored**. Because the iPad surface is so much bigger than the iPhone's, at any given time the chance of focusing close enough to the tab bar so we actually notice it is quite small, particularly since the most important information is often toward the top of the page.



The Steering Wheel

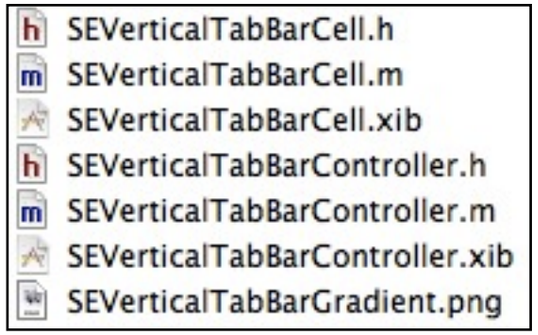
SEVertical TabBar Controller

API

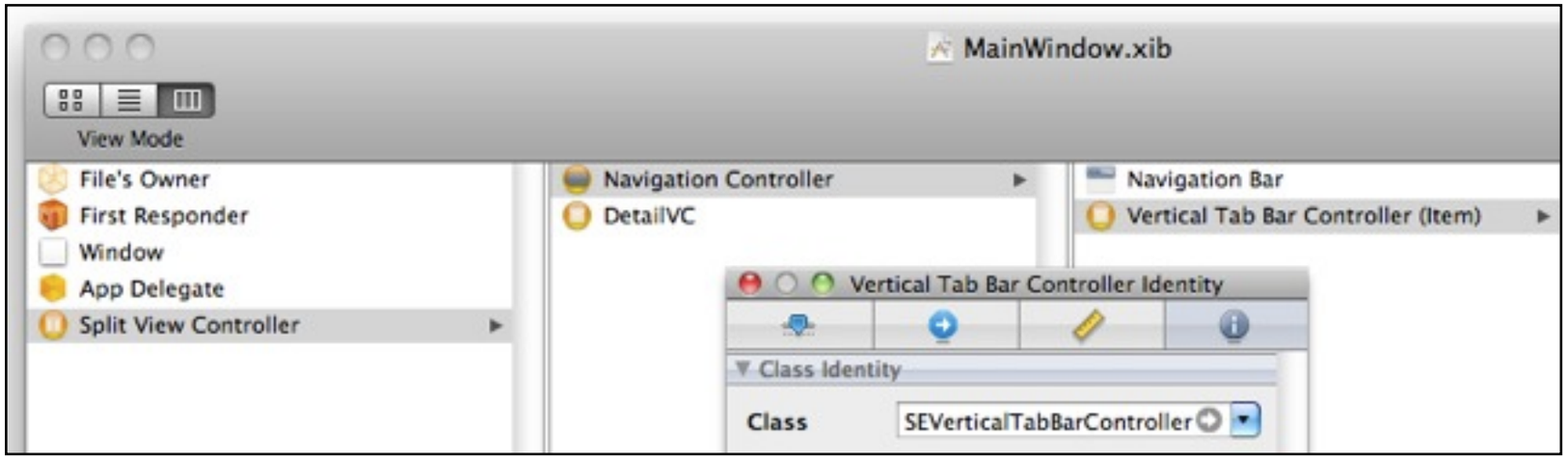
```
@protocol SEVerticalTabBarControllerDelegate
- (BOOL)    tabBarController:(SEVerticalTabBarController *)vtbc
  shouldSelectViewController:(UIViewController *)viewController;
- (void)    tabBarController:(SEVerticalTabBarController *)vtbc
     didSelectViewController:(UIViewController *)viewController;
- (void)    tabBarController:(SEVerticalTabBarController *)vtbc
   didUnselectViewController:(UIViewController *)viewController;
@end

// To change the detail view controller, post kSEChangeDetailViewControllerNotification
// notification with the view controller to display as the notification object, such as:
// [[NSNotificationCenter defaultCenter]
postNotificationName:kSEChangeDetailViewControllerNotification object:detailVC userInfo:nil];

@interface SEVerticalTabBarController : UIViewController <UITableViewDataSource,
UITableViewDelegate> {
    // ...
}

@property (nonatomic, retain) UITableView *tabBarTableView;
@property (nonatomic, retain) UIViewController *selectedViewController;
@property (nonatomic, retain) NSArray *viewControllers;
@property (nonatomic, assign) id <SEVerticalTabBarControllerDelegate>
                                  verticalTabBarControllerDelegate;
@property (nonatomic, assign) UISplitViewController *splitViewController;
```

Integration

1. download the files



2. tell your split view controller about it



Integration

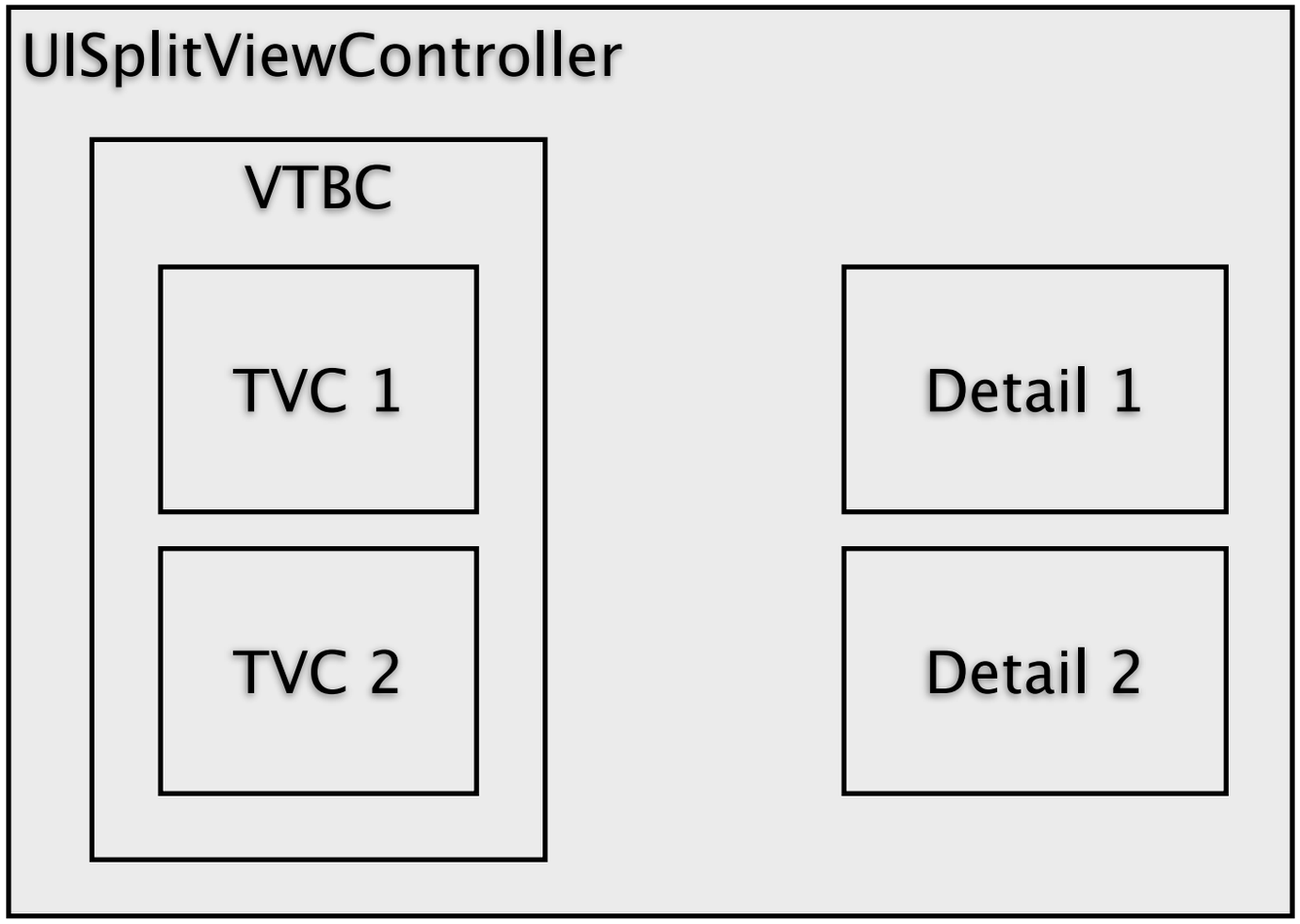
3. setup the view controllers

```
NSMutableArray *viewControllers = [NSMutableArray array];

MasterTVC *masterTVC =
    [[[MasterTVC alloc] initWithNibName:@"MasterTVC" bundle:nil] autorelease];
masterTVC.title = @"Countries";
masterTVC.items = [NSArray arrayWithObjects:@"France", @"UK", nil];
[viewControllers addObject:masterTVC];

verticalTabBarController.viewControllers = viewControllers;
```

4. post notifications when detail changes

```
detailVC.representedObject = o;

[[NSNotificationCenter defaultCenter]
    postNotificationName:kSEChangeDetailViewControllerNotification
                  object:detailVC
                userInfo:nil];
```

Views Structure



TabBar implementation

```
@interface SEVerticalTabBarController : UIViewController <UITableViewDataSource, UITableViewDelegate> {
    // ...
}
```

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

   [selectedViewController viewWillDisappear:NO];
   [selectedViewController.view removeFromSuperview];
   [selectedViewController viewDidDisappear:NO];

   self.selectedViewController = [viewControllers objectAtIndex:indexPath.row];

   [verticalTabBarControllerDelegate tabBarController:self didUnselectViewController:selectedViewController];

   CGRect rect = CGRectMake(tabBarTableView.bounds.size.width,
                            0,
                            self.view.bounds.size.width - tabBarTableView.bounds.size.width,
                            self.view.bounds.size.height);

   selectedViewController.view.frame = rect;

   [self.view addSubview:selectedViewController.view];
   [selectedViewController viewWillAppear:NO];
   [selectedViewController viewDidAppear:NO];

   self.title = selectedViewController.title;

   [verticalTabBarControllerDelegate tabBarController:self didSelectViewController:selectedViewController];
}
```

Cells Implementation

```
+ (CAGradientLayer *)gradientLayerWithBounds:(CGRect)rect {
   // same gradient as Apple's UITabBar
   UIColor *backgroundColor = [UIColor colorWithRed:37/255. green:37/255. blue:37/255. alpha:1.0];
   UIColor *inColor = [UIColor colorWithRed:54/255. green:54/255. blue:54/255. alpha:1.0];
   UIColor *outColor = [UIColor colorWithRed:73/255. green:73/255. blue:73/255. alpha:1.0];

   CAGradientLayer *gl = [CAGradientLayer layer];
   gl.frame = rect;
   gl.backgroundColor = [backgroundColor CGColor];
   gl.colors = [NSArray arrayWithObjects:
                 (id)[backgroundColor CGColor],
                 (id)[backgroundColor CGColor],
                 (id)[inColor CGColor],
                 (id)[outColor CGColor], nil];
   gl.locations = [NSArray arrayWithObjects:
                    [NSNumber numberWithDouble:0.0],
                    [NSNumber numberWithDouble:0.5],
                    [NSNumber numberWithDouble:0.5],
                    [NSNumber numberWithDouble:1.0], nil];
   gl.startPoint = CGPointMake(0.0, 0.5);
   gl.endPoint   = CGPointMake(1.0, 0.5);
   gl.masksToBounds = YES;
   gl.cornerRadius = 5.0;

   return gl;
}
```

Cells Implementation

```
- (void)setSelected:(BOOL)selected animated:(BOOL)animated {
   // ...

   NSString *title = [representedViewController.title lowercaseString];
   NSString *imageName = selected ? [title stringByAppendingString:@"_selected"] : title;
   imageName = [imageName stringByAppendingPathExtension:@"png"];

   if(selected) {
         if(!gradientLayer) {
             CGFloat xOffset = 6.0;
             CGFloat yOffset = 15.0;
             CGRect rect = CGRectMake(...);
             self.gradientLayer = [SEVerticalTabBarCell gradientLayerWithBounds:rect];
         }

         [self.layer insertSublayer:gradientLayer atIndex:0];
   } else {
         [gradientLayer removeFromSuperlayer];

         self.layer.backgroundColor = NULL;
         self.layer.masksToBounds = NO;
         self.layer.cornerRadius = 0.0;
   }

   imageView.image = [UIImage imageNamed:imageName];
}
```

Advantages

- Users can use their thumbs
- Bonus: you don’t have to use blue gradients
- You can use any image you want indeed

Tradeoffs

- No alpha composition
- UI would becomes hazardous in portrait
- App Store reviewers unfamilar…

… yet!

Can I download if for free?

http://github.com/nst/SEVerticalTabBar

Thank you, and happy coding!