

Layout Transitions

Aleksander Zubala

@alekzubala

Berlin, 2015

Transitions

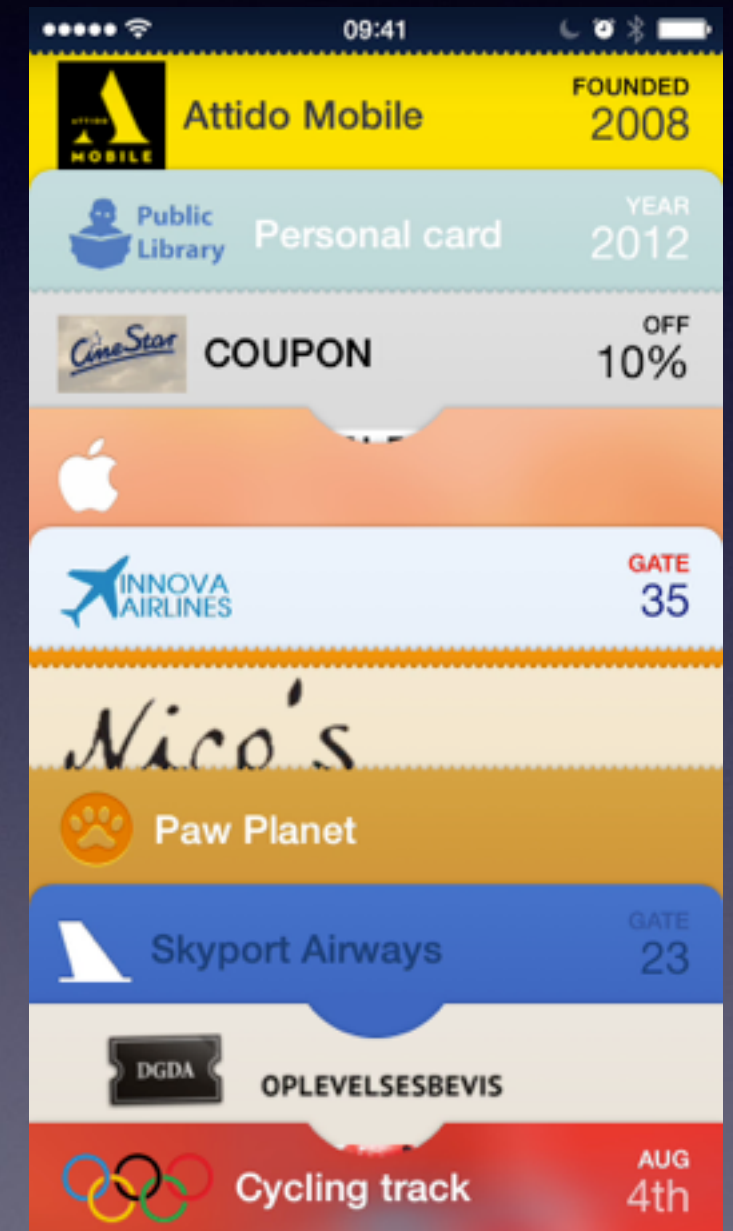
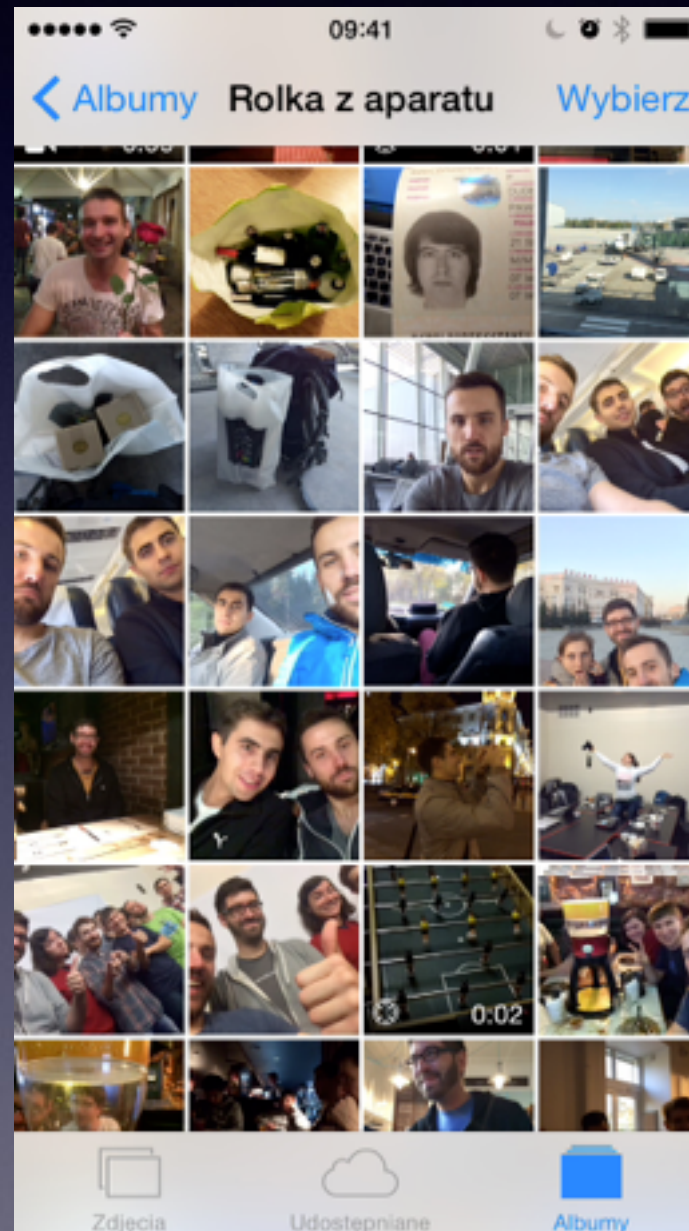
Why bother, anyway?

- Change the context (push, modal)
- Change presentation (details)
- Show off :)

What's good transition?

- not slow, not fast - **timing**
- subtle, balanced between flashy and practical - **appeal**
- presenting main idea clearly - **staging**

Examples



Automatic

- transition between layouts that can be interpolated
- change layout on collection view with `setLayout:animated:completion`
- non-interactive
- no control over the timing

Automatic

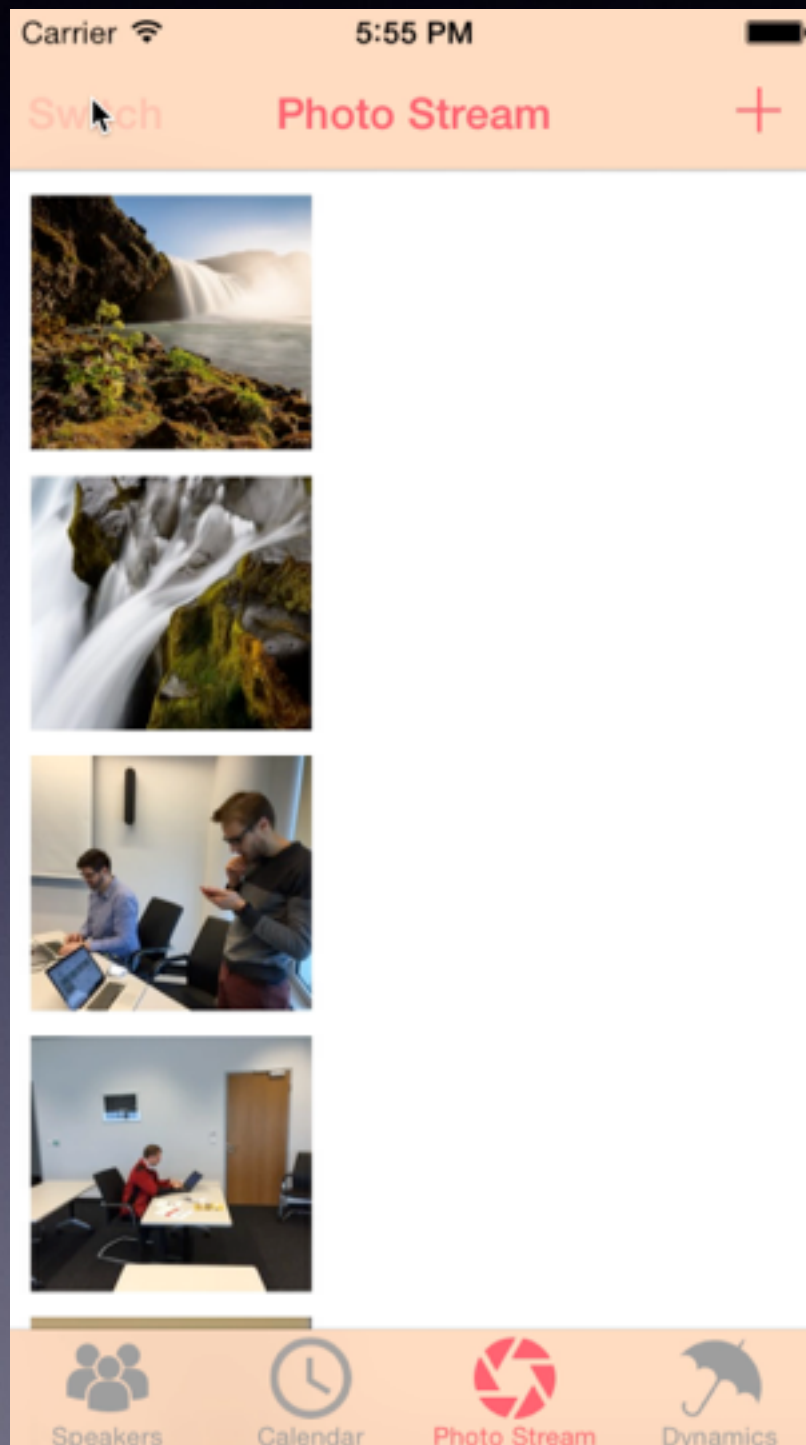
- How about we use this info?
(UICollectionViewLayout.h):

```
// This set of methods is called when the collection view undergoes an  
animated transition such as a batch update block or an animated bounds  
change.  
// For each element on screen before the invalidation,  
finalLayoutAttributesForDisappearingXXX will be called and an animation  
setup from what is on screen to those final attributes.  
// For each element on screen after the invalidation,  
initialLayoutAttributesForAppearingXXX will be called an an animation setup  
from those initial attributes to what ends up on screen.
```


Task 1: Final/Initial

- `git checkout lt-task-1-setup`
- Group named `Photo Stream`
- Key classes:
 - `PhotoStreamViewController`
 - `LeftLayout`
 - `RightLayout`
- Automatic transition (as expected) produces interpolation between left and right layout

Task 1: Final/Initial



TODOs:

- In `LeftLayout` adjust layout attributes in `initial/final` methods so items are pushed outside the screen to the left
- The same goes for `RightLayout`, but push items outside the screen to the right
- `git checkout lt-task-1-solution`

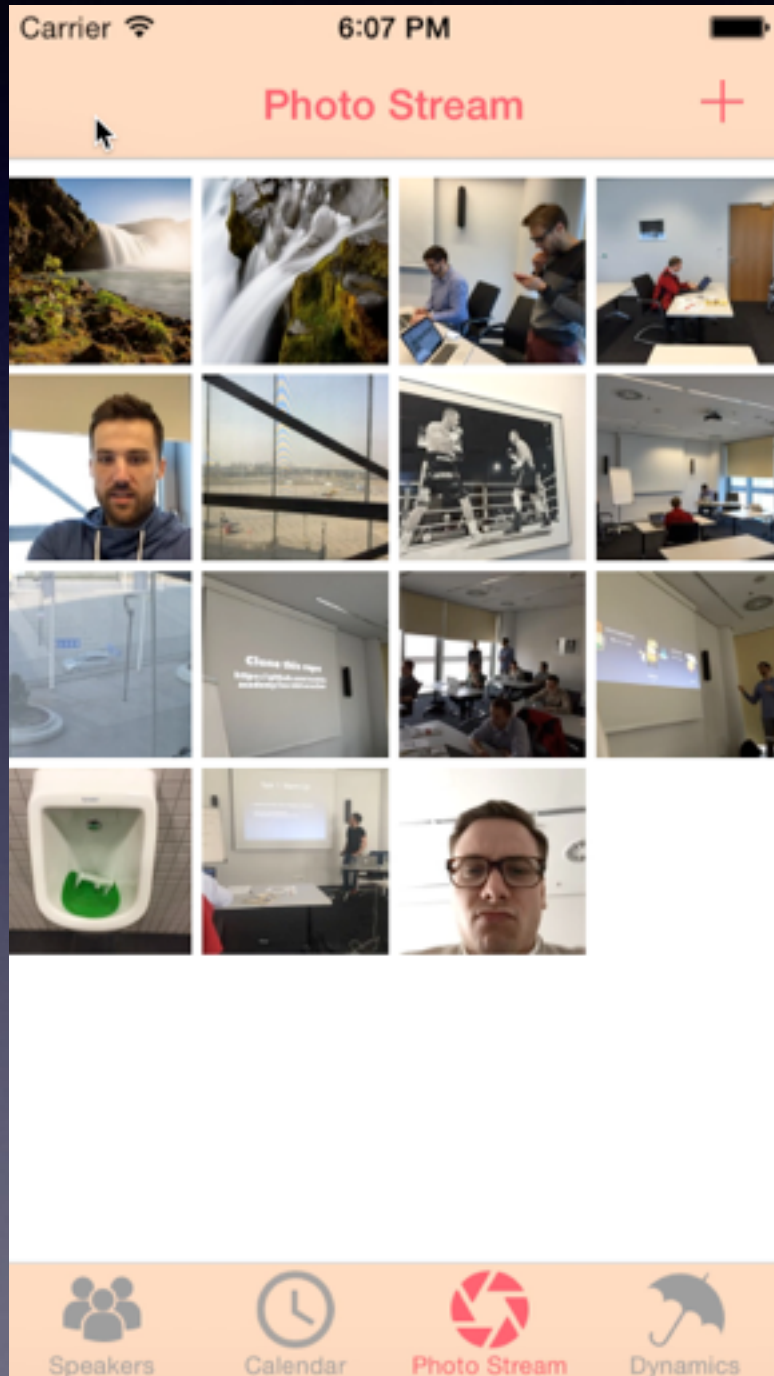
Layout to Layout

- Property on `UICollectionViewController` called `useLayoutToLayoutNavigationTransitions`
- Transition using navigation controller
- Collection view controller as the root object
- Perform transition by setting property to **YES** on controller which will be pushed
- Interactive for free!

Task 2: Layout to Layout

- `git checkout lt-task-2-setup`
- Group named `Photo Stream`
- Key classes:
 - `PhotoStreamLayout`
 - `PhotoStreamViewController`
 - `StreamItemPreviewLayout`
 - `StreamItemPreviewViewController`

Task 2.1: Layout to Layout

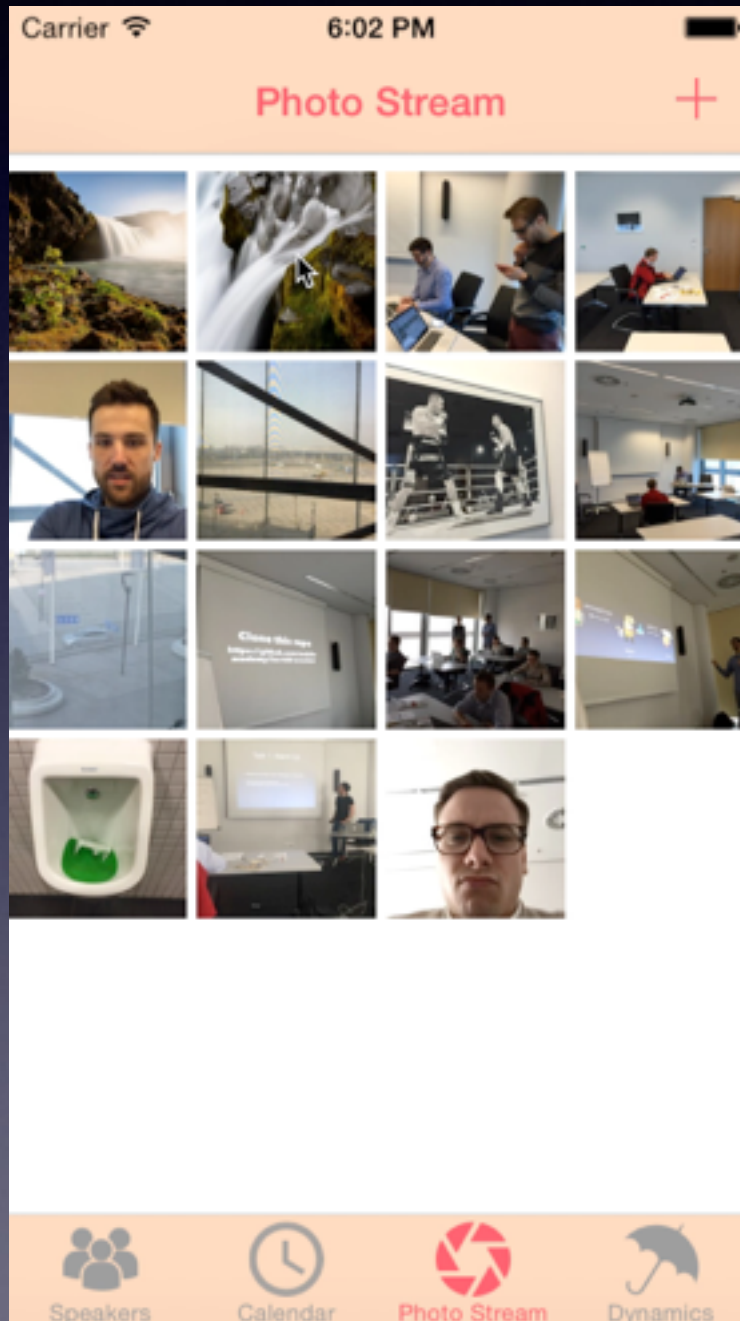


TODO:

- Push (instead of present modal) `StreamItemPreviewViewController` when cell is pressed on `PhotoStreamViewController`
- Make sure to set `useLayoutToLayoutNavigationTransitions` to YES on `StreamItemPreviewViewController` before pushing onto stack
- `git checkout lt-task-2.1-solution`

Home Assignment

Task 2.2: Layout to Layout



TODO:

- In `StreamItemPreviewLayout` set item size to full screen only for selected item, other items should have zero size
- Set `alpha` to 0 for not selected items
- Use initial/final layout hooks (see Task 1) to control transition
 - `PhotoStreamLayout`: initial attributes for appearing, should set full screen size for selected item
 - `StreamItemPreviewLayout`: final attributes for disappearing set grid item size for all items but selected
- Selected index path needs to be tracked in both layouts
- `git checkout lt-task-2.2-solution`

Task 2: Layout to Layout

- Easy but automatic transition
- Interactive for free
- Presented controller is 'shallow'
 - No delegate
 - No data source
 - Shared collection view

Automatic is boring ;)

- So far all transitions automatic
- How about custom stuff?
- Interactive is the new black
- New API from iOS 7 in `UICollectionView`

`startInteractiveTransitionToCollectionViewLayout:completion:`

Interactive Transition

- Change the layout using an intermediate transition
- Need to setup gesture or other touch-event handling code
- Update `transitionProgress`
- To finish: `finishInteractiveTransition`
- To cancel: `cancelInteractiveTransition`

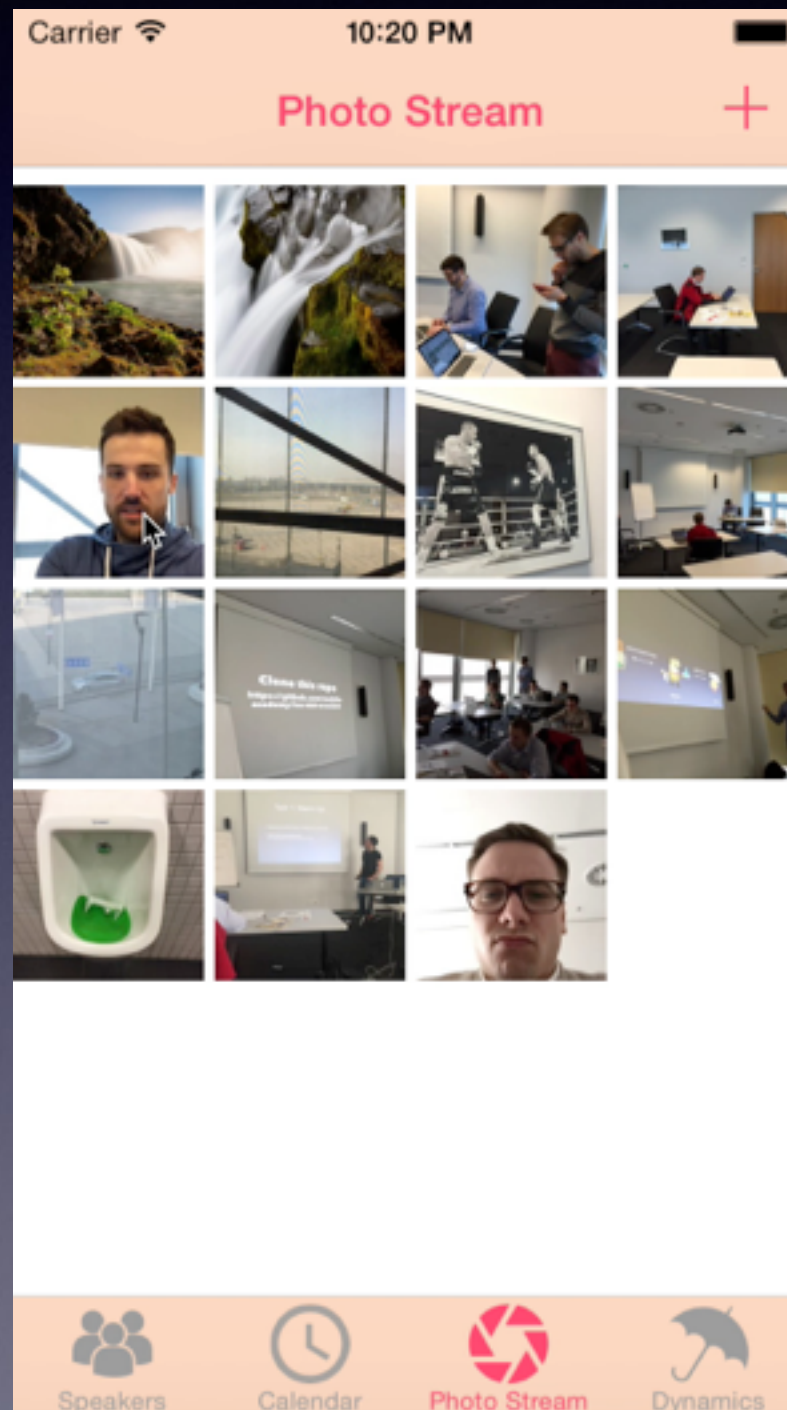
Interactive Transition

- `UICollectionViewTransitionLayout` class by default
- Custom transition object instead, implement the `collectionView:transitionLayoutForOldLayout:newLayout:`

Task 3: Custom Interactive Transition

- `git checkout lt-task-3-setup`
- Group named Photo Stream
- Key classes:
 - `PhotoStreamViewController`

Task 3: Custom Interactive Transition



Task 3.1: Custom Interactive Transition

- Implement pinch callback:
 - Began - select index path, start transition
 - Changed - update progress, invalidate layout
 - Ended - finish transition or cancel depending on the progress
 - Cancelled - cancel transition
- `git checkout lt-task-3.1-solution`

Task 3.2: Custom Interactive Transition

- Implement `collectionView:transitionLayoutForOldLayout:newLayout:`
- Create subclass of `UICollectionViewTransitionLayout`
 - `layoutAttributesForElementsInRect:`
 - `layoutAttributesForItemAtIndexPath`
- Make use of `transitionProgress`
- `git checkout lt-task-3.2-solution`

- It-task-3.2-setup

Where to go next?

- WWDC 2013, Session 218: Custom Transitions Using View Controllers
- objc.io: View Controller Transitions, Issue #5 iOS 7, October 2013
- “Custom UIViewController Transitions” - Ash Furrow (teehanlax.com/blog)
- UIKonf 2014 - Eric Allam: Building better transitions

Thanks!