Mobile Application Programming

Xcode Tools
The Iterative Design Process
Final Projects

- A Final Project is:
  - Large (~40 Hrs / Person)
  - Real (Meets Actual Need)
  - Complete (Feature-wise)
  - Polished (Store Quality)
  - Valuable (Someone Would Like to Buy It)

- 1 Programmer Examples
  - Photo App with Filters
  - Networked Social App
  - Asteroids with Pickups
  - Battleship (2 iDevices)
  - Simple OpenGL Game
Editor - Inspector

```swift
import UIKit

class StackListViewController: UIViewController, UICollectionViewDataSource, UICollectionViewDelegate, StackViewControllerDelegate {
    // MARK: - Member Data
    private var _cardLibrary: CardLibrary = CardLibrary()
    private var _normalItems: [UIBarButtonItem] = []
    private var _editingItems: [UIBarButtonItem] = []

    // MARK: - UICollectionView Overrides
    override func viewDidLoad() {
        super.viewDidLoad()
        let flowLayout = UICollectionViewFlowLayout(
            collectionView: collectionView,
            frame: CGRectZero, 
            collectionViewLayout: flowLayout
        )
        override func viewDidLoad() {
            super.viewDidLoad()
            // Create the set of toolbar items that are needed when in normal mode and editing mode
            _normalItems = [
                UIBarButtonItem(barButtonSystemItem: .Add, target: self, action: "addPressed")!,
                UIBarButtonItem(title: "Edit", style: .Plain, target: self, action: "editPressed")
            ]
            _editingItems = [
                UIBarButtonItem(barButtonSystemItem: .Cancel, target: self, action: "cancelPressed")!,
                UIBarButtonItem(barButtonSystemItem: .Trash, target: self, action: "deletePressed")!
            ]

            // Set the normal mode initially
            navigationItem.rightBarButtonItem = _normalItems[0]
            // Setup the stack list view to show previews of the card stacks
            stackListView.backgroundColor = UIColor.grayColor
            stackListView.dataSource = self
            stackListView.delegate = self
            stackListView.registerClass(UICollectionViewCell.self, forCellWithReuseIdentifier: NSStringFromClass(UICollectionViewCell.self))
            stackListViewLayout.sectionInset = UIEdgeInsetsMake(0, 0, 0, 0)
            stackListViewLayout.minimumInteritemSpacing = 8
        }

        override func willAppear(animated: Bool) {
            super.viewDidLoad()
            cardLibrary.removeEmptyStacks()
        }
    }

    // MARK: - UICollectionViewDataSource
    func collectionView(_ collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int {
        return _normalItems.count
    }

    func collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath) -> UICollectionViewCell {
        let cell = collectionView.dequeueReusableCell(withReuseIdentifier: "cell", forIndexPath: indexPath) as! CardViewCell
        cell.titleLabel.text = "Stack Title"
        return cell
    }

    // MARK: - UICollectionViewDelegate
    func collectionView(_ collectionView: UICollectionView, didSelectItemAt indexPath: IndexPath) {
        // Handle selection
    }

    // MARK: - StackViewControllerDelegate
    func numberOfSections(in collectionView: UICollectionView) -> Int {
        return 1
    }

    func collectionView(_ collectionView: UICollectionView, viewForSupplementaryElementOfSection section: Int, kind: ReuseIdentifier) -> ReusableCell {
        let cell = collectionView.dequeueReusableSupplementaryView(
            kind: kind, 
            for: indexPath
        ) as! SupplementaryCell
        return cell
    }
}
```
Editor - Visual Differences
Instruments
Instruments
Interface Builder
Interface Builder

- Provides **Storyboard and XIB files** to make the creation of composite views a WYSIWYG experience.
- XIB files contain a **freeze-dried full-screen UI** that is loaded into memory along with a view controller, filling in **IBOutlet vars** with object instances.
- Storyboards contain many full-screen views and the relationships between them, called **segues**.
Interface Builder