iPhone Programmers Association

iPhone Software Architecture - Matt Stoker
Design and Implementation

So, where does everything go?

- Apple Model-View-Controller program design
- Typical Control Flow - Entry Point, Delegates, etc.
- Real World Example - iJezzball
Model View Controller (MVC)

Controller = Adapter
Model View Controller (MVC)

Controller = Adapter
Typical Control Flow

Application Entry Point

```c
int main(int argc, char *argv[]) {
    // Create autorelease pool and instantiate app and app delegate
    NSAutoreleasePool* pool = [[NSAutoreleasePool alloc] init];
    int retVal = UIApplicationMain(argc, argv, nil, @"MSAppDelegate");
    [pool release];
    return retVal;
}
```
Typical Control Flow

GUI Entry Point

- (void) applicationDidFinishLaunching:(UIApplication*)application
{
    //Create window
    _window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];

    //Create and setup main view controller
    _mainViewController = [[MainViewController alloc] init];
    [_window addSubview: _mainViewController.view];

    //Show window
    [_window makeKeyAndVisible];
}
Typical Control Flow

Controller Structure

@interface MainViewController : UIViewController
{
    MainView* _contentView;
}

- (id)init;
- (void)dealloc;
- (void)loadView;
@end
Typical Control Flow

View Structure

@protocol MainViewDelegate <NSObject>
@required - (void) viewNeedsSomeValue:(MainView*)view;
@optional - (void) userPerformedSomeActionOnView:(MainView*)view;
@optional - (void) userPerformedSomeOtherActionOnView:(MainView*)view;
@end

@interface MainView : NSObject
{
    UIImageView* _someImage;
    UIButton* _someButton;
    UILabel* _someDynamicLabel;
    id<MainViewDelegate> _delegate;
    float _viewProperty;
}
@end

- (id) initWithFrame:(CGRect)frame;
- (void) dealloc;

@property (assign) id<MainViewDelegate> delegate;
@property (assign) float viewProperty;

- (void) doSomethingWithPoint:(CGPoint)point;
- (void) doSomethingWithObject:(id)object;
@end
Typical Control Flow

Model Structure

@protocol EngineDelegate <NSObject>
@required - (void) engineNeedsSomeValue:(Engine*)engine;
@optional - (void) userPerformedSomeActionOnView:(Engine*)engine;
@optional - (void) userPerformedSomeOtherActionOnView:(Engine*)engine;
@end

@interface Engine : NSObject
{
    id<EngineDelegate> _delegate;
    NSString* _engineProperty0;
    float _engineProperty1;
}
-	(id) initWithDataFile:(NSString*)dataFile;
-	(void) dealloc;
@property (assign) id<MainViewDelegate> delegate;
@property (retain) NSString* engineProperty0;
@property (assign) float engineProperty1;
-	(void) doSomethingWithValue:(int)value;
-	(void) doSomethingWithObject:(id)object;
@end
Real World Example

- Real world programs are usually not perfectly formed.
- Good real world programs do follow some model, though, or they turn into unmaintainable, spaghetti code garbage.
- Make your objects as reusable as you can, as you will definitely reuse some of them, even when you think you wouldn’t.
If You Don’t Want to Suck...

Write some code every day!

Even if it is only for 20 minutes
If You Don’t Want to Suck...

Start writing some code right now!

Literally. Really. Immediately. Open Xcode...
Suggestions

- Fairly Easy
- Connect Four
- Oh-Mok
- Tic Tac Toe
- Nim
- Mastermind
Suggestions

- Not As Easy Games
- Battleship
- Hangman
- Boggle
- Yahtzee
- Blackjack
- Tetris