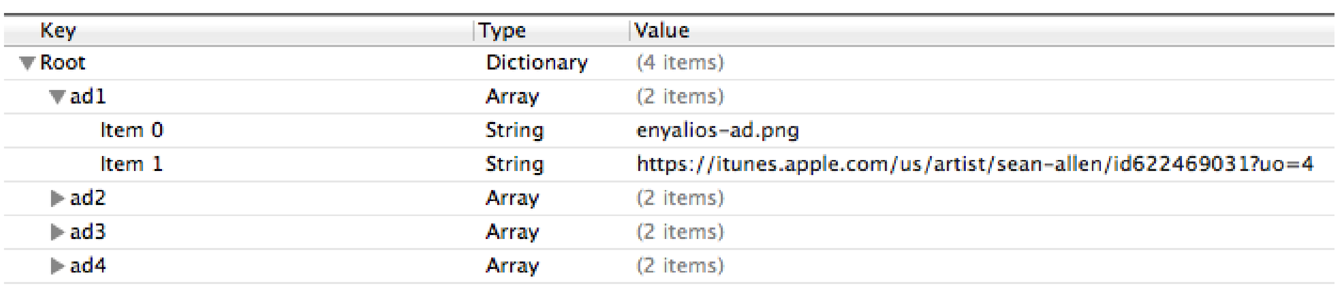
Overview

EEAdBannerView is an image view object that will cycle ads provided by the user. Currently, EEAdBannerView supports iPhone and iPod Touch portrait mode banner ads only.

Plist

EEAdBannerView retrieves the ad information from the EEAdBanners.plist. Format of the information found in EEAdBanners.plist is as follows:



Key	Type	Value
▼Root	Dictionary	(4 items)
▼ad1	Array	(2 items)
Item 0	String	enyalios-ad.png
Item 1	String	https://itunes.apple.com/us/artist/sean-allen/id622469031?uo=4
▶ad2	Array	(2 items)
▶ad3	Array	(2 items)
▶ad4	Array	(2 items)

The key for each ad array should be of the form 'ad#' starting at 1 and incrementing up. The first element of the array is a string with the image file and the second element of the array is a string with the url link for the ad. The url string can be set to nil.

Images

The ad images provided can be of any type supported by UIImageView and should be of the following sizes:

Device and orientation	Dimensions	Naming convention
iPhone/iPod portrait non-retina	320x50	image.png
iPhone/iPod portrait retina	640x100	image@2x.png
iPhone/iPod landscape non-retina	Not available	Not available
iPhone/iPod landscape retina	Not available	Not available
iPhone 5 landscape non-retina	Not available	Not available
iPhone 5 landscape retina	Not available	Not available

Properties

adBannerDisplayTime
a double value that determines how long an ad will show before displaying the next ad.
@property(nonatomic) double adBannerDisplayTime

adBannerCycleMode
a boolean value that determines what order the ads should be displayed.
@property(nonatomic) BOOL adBannerCycleMode
Available options are *kAdBannerCycleModeNormal* and *kAdBannerCycleModeRandom*

Tasks

init
initializes the adBannerView. It should be noted that the banner is sized and located just off the bottom of the screen. The user must call *loadAdBanner* to see the banner.
- (id)init

loadAdBanner
animates the banner from of screen to the bottom of the screen for the user to view.
- (void)loadAdBanner

unloadAdBanner
animates the banner off screen and stops cycling ads.
- (void)unloadAdBanner

Integration With iAd

This class was designed to fill in when iAd fails to load an ad. To get this to work call loadAdBanner when iAd fails and call unloadAdBanner when iAd loads an ad:

```
- (void)bannerViewDidLoadAd:(ADBannerView *)banner
{
    // Move EEAdBannerView off screen
    [myAdBanner unloadAdBanner];

    // Move iAd onscreen
    [UIView animateWithDuration:0.5
                 animations:^{ // Move iAds ON SCREEN
                     [iAdView setFrame:CGRectMake(0, [UIScreen mainScreen].bounds.size.height - 50, 320, 50)];
                 }];
}

- (void)bannerView:(ADBannerView *)banner didFailToReceiveAdWithError:(NSError *)error
{
    // Move iAd off screen
    [UIView animateWithDuration:0.5
                 animations:^{
                     [iAdView setFrame:CGRectMake(0, [UIScreen mainScreen].bounds.size.height - 20, 320, 50)];
                 }
                 completion:^(BOOL finished){
                     // Move EEAdBannerView on screen
                     [myAdView loadAdBanner]
                 }];
}
```

Integration with Google Analytics

EEAdBannerView is setup to report taps on your banner ad to your google analytics account. Your application must already have google analytics setup. To enable ad tracking there are 4 lines to uncomment:

In EEAdBannerView.h
#import "GAI.h"
id<GAITracker> tracker;

In EEAdBannerView.m
tracker = [[GAI sharedInstance] defaultTracker];
[tracker sendEventWithCategory:@"uiAction" withAction:@"adTap" withLabel:[[adBannerDictionary objectForKey:key] objectAtIndex:0] withValue:nil];

The tap will send the name of your ad's image file, so it is a good idea to name them something descriptive like killerPhotoApp.png

In Development

- Support for more devices and orientations
- Dynamic loading of ads so developers do not need to submit an update to the app store to change, add, or update their ads
- Full screen ads for developers who want to piss of their users!