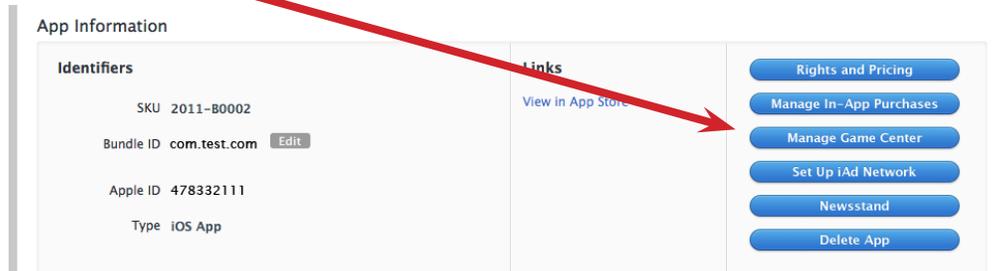


Integrating Game Center into a BuzzTouch 1.5 app

This tutorial assumes you have created your app and downloaded the source code; created an App ID in the iOS Provisioning Portal, and registered your app in iTunes Connect.

Step 1. Creating the Leaderboard in iTunes Connect

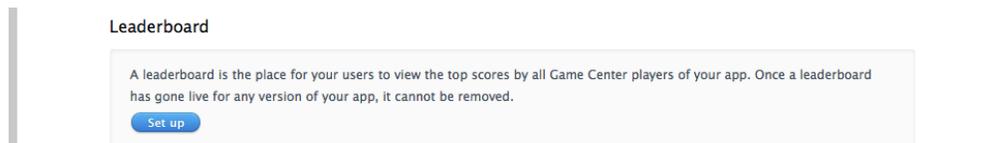
Login into your iTunes Connect account and select your app. From the app management screen click on “Manage Game Center.”



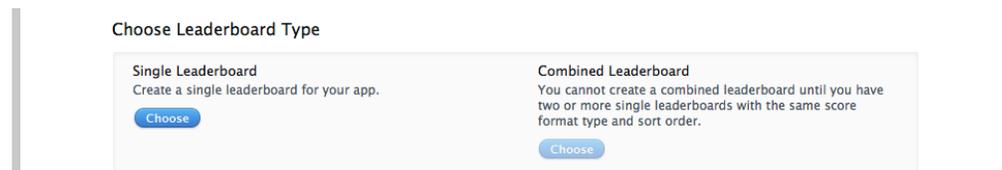
On the next screen, click to enable Game Center.



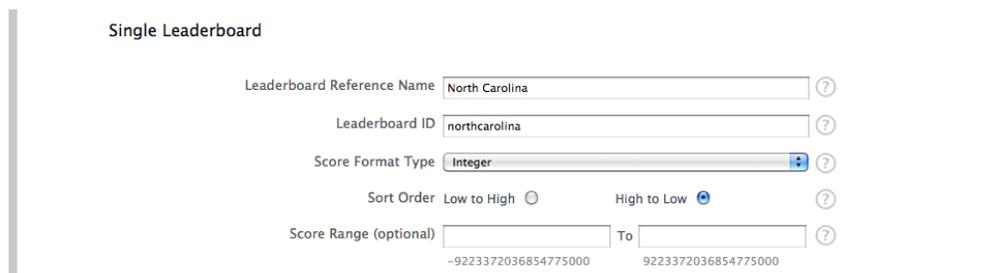
Once enabled, click on “Set up” under Leaderboard.



Select “Choose” under Single Leaderboard.



Name your Leaderboard and give it a Leaderboard ID. Remember the ID. It is what you will reference to connect your app to the Game Center. Select “Integer” for Format type (or whatever you want) and select a sort order.

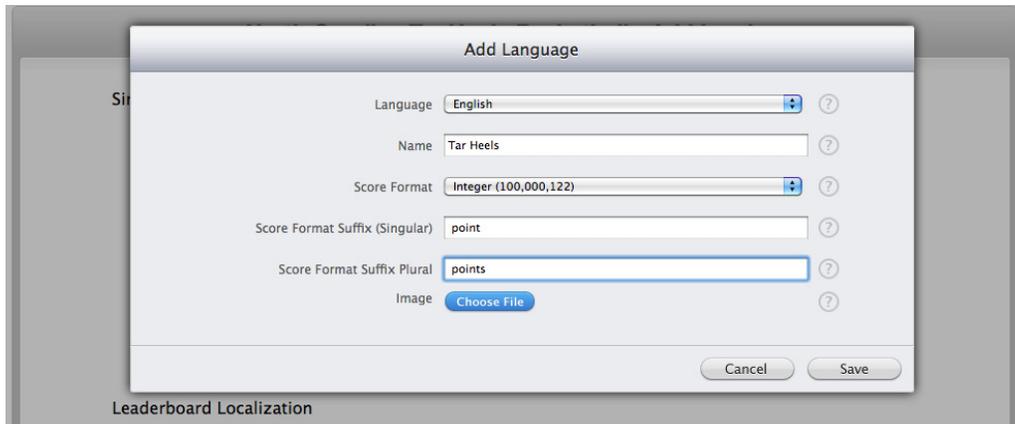


Integrating Game Center

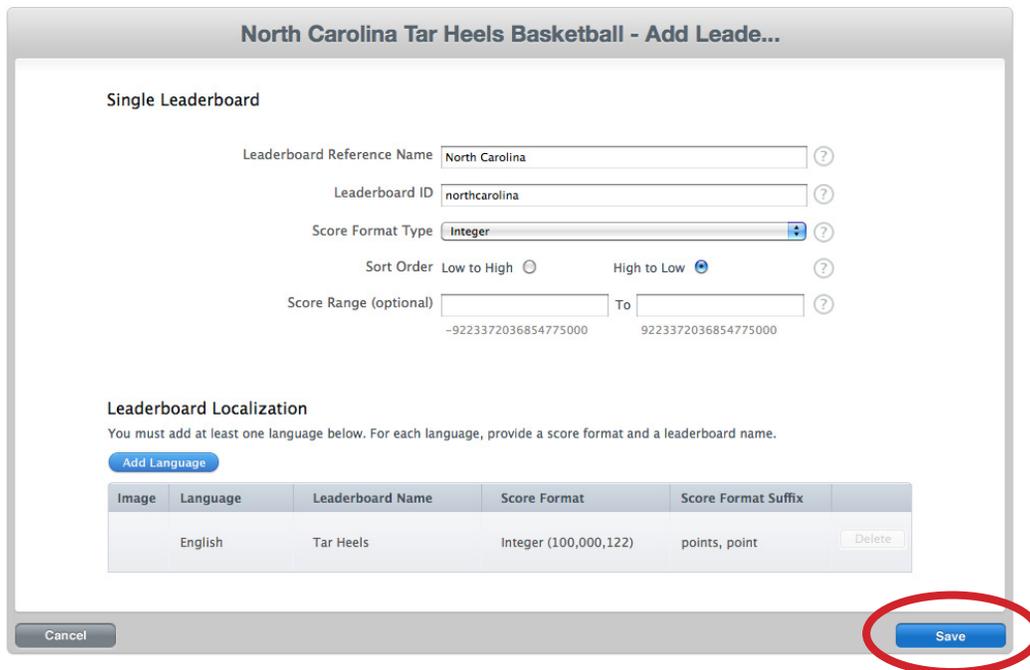
Click “Add Language” and a new box will open. Select your language and enter a Name. This will be the name that appears in the title bar of the Game Center.

Select a Score Format and enter a suffix. Then upload an image for your Leaderboard. This image must be 512 x 512 px like the app’s icon. You can use the app’s icon here if you’d like.

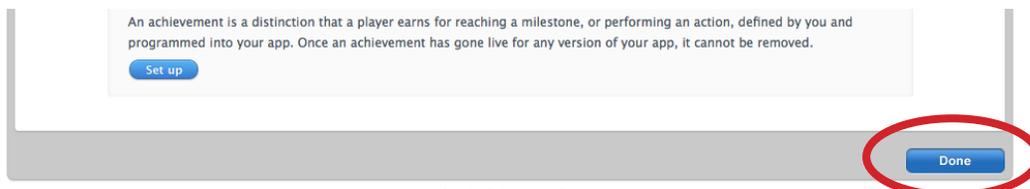
Click “Save” when done.



Follow the same steps to add additional language localizations if you’d like. When you’re finished, click “Save.”

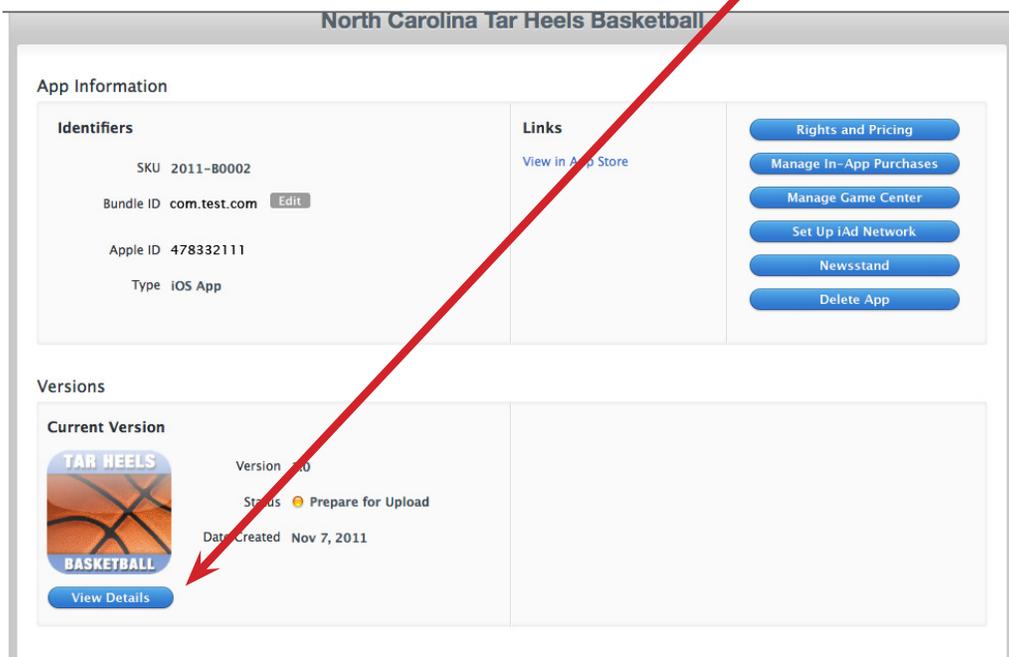


At the Game Center screen, click “Done” to finish this first step.

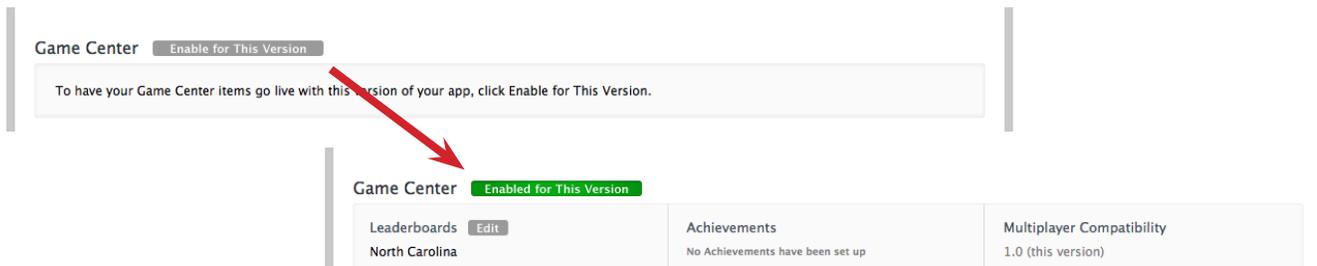


Integrating Game Center

The final step is to enable Game Center in the app. Click the “View Details” button.



Click on the “Enable for This Version” button. Once enabled, it will turn green.



Once done, you can sign out of iTunes Connect.

2. Download Apple’s sample code.

You’ll need to download the GK Tapper sample code from Apple.

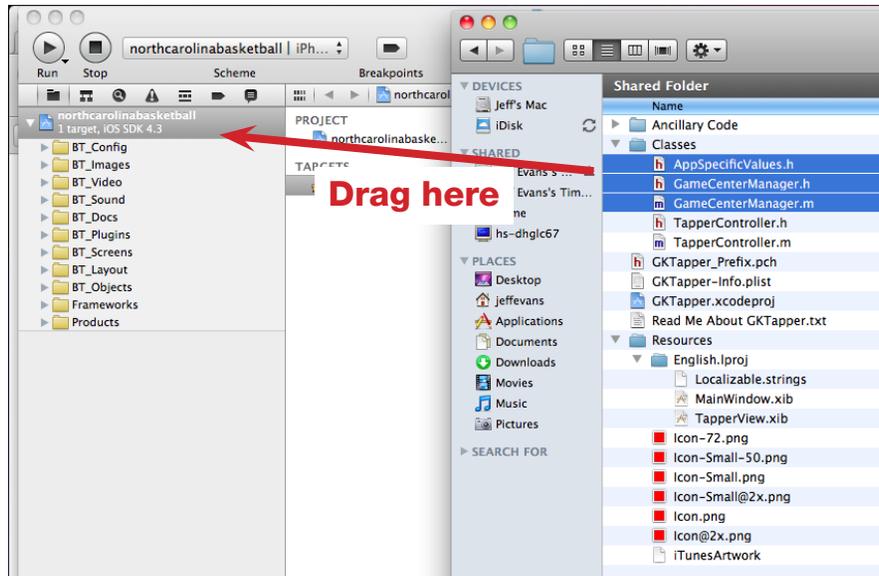
<http://developer.apple.com/library/ios/#samplecode/GKTapper/Introduction/Intro.html>

Unzip the file. You’ll need several of the code files in the next step.

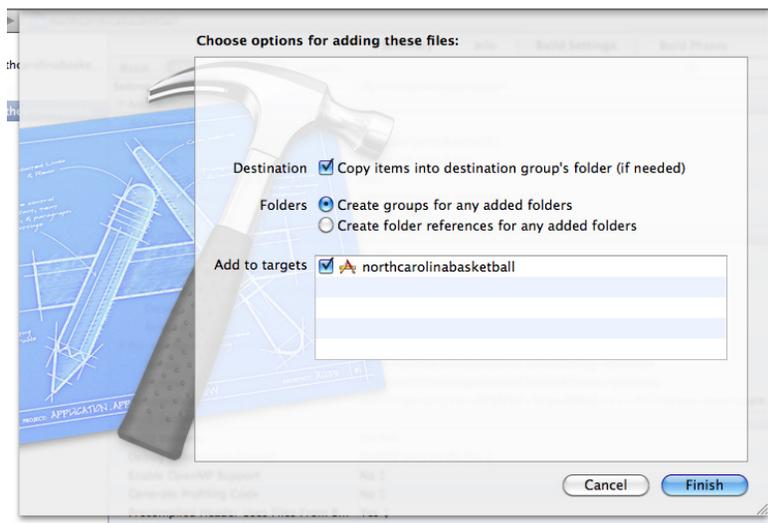
Integrating Game Center

3. Open your app in xCode.

Open your app in xCode, then drag the following three files into your project: AppSpecificValues.h, GameCenterManager.h and GameCenterManager.m



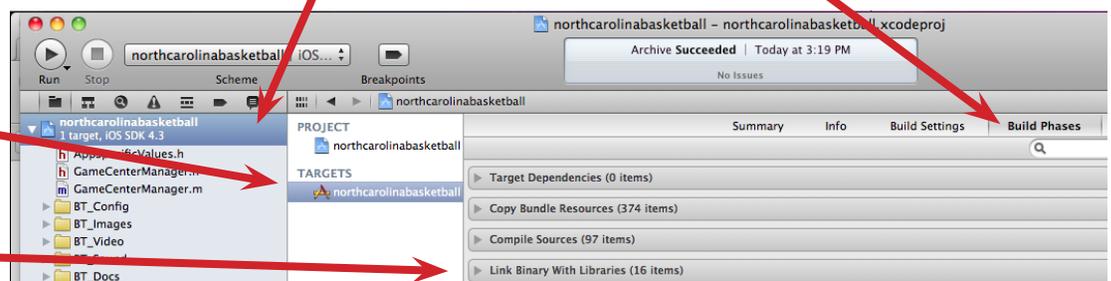
Make sure to check “Copy items into destination group’s folder.” Then click “Finish.”



Add the GameKit framework to your project. a) Click your project; b) Click “Build Phases”;

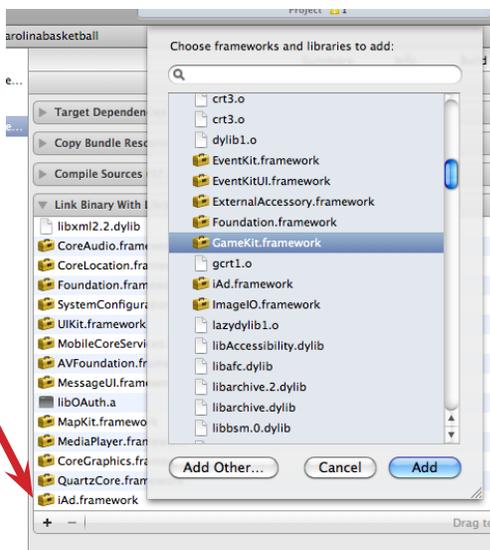
c) Click your app under “Targets”;

d) Click the arrow next to “Link Binary With Libraries.”



Integrating Game Center

Click the “+” button to open the add frameworks dialogue box.



Then select the GameKit framework and click “Add.”

4. Select the “BT_viewController.h” file in the “BT_Layout” folder.



Add the following below the list of “#import” statements:

```
#import <GameKit/GameKit.h>
#import "GameCenterManager.h"
@class GameCenterManager;
```



Then add the following to the “@interface” statement so it looks like this.

```
, ADBannerViewDelegate, UIActionSheetDelegate, GKLeaderboardViewControllerDelegate,
GKAchievementViewControllerDelegate, GameCenterManagerDelegate
```

Add the following code above the “BT_irem *screenData;” line.

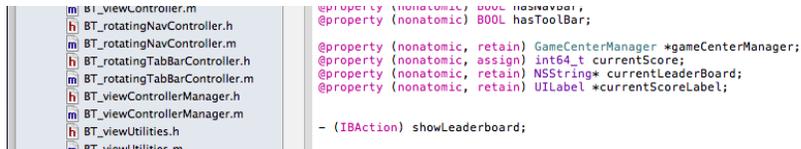
```
GameCenterManager *gameCenterManager;
int64_t currentScore;
NSString* currentLeaderBoard;
IBOutlet UILabel *currentScoreLabel;
```

Integrating Game Center

Add the following to the file at the bottom of the list of “@property” statements:

```
@property (nonatomic, retain) GameCenterManager *gameCenterManager;
@property (nonatomic, assign) int64_t currentScore;
@property (nonatomic, retain) NSString* currentLeaderBoard;
@property (nonatomic, retain) UILabel *currentScoreLabel;

- (IBAction) showLeaderboard;
```



```
m BT_viewController.m
h BT_rotatingNavController.h
m BT_rotatingNavController.m
h BT_rotatingTabBarController.h
m BT_rotatingTabBarController.m
h BT_viewControllerManager.h
m BT_viewControllerManager.m
h BT_viewUtilities.h
m BT_viewUtilities.m

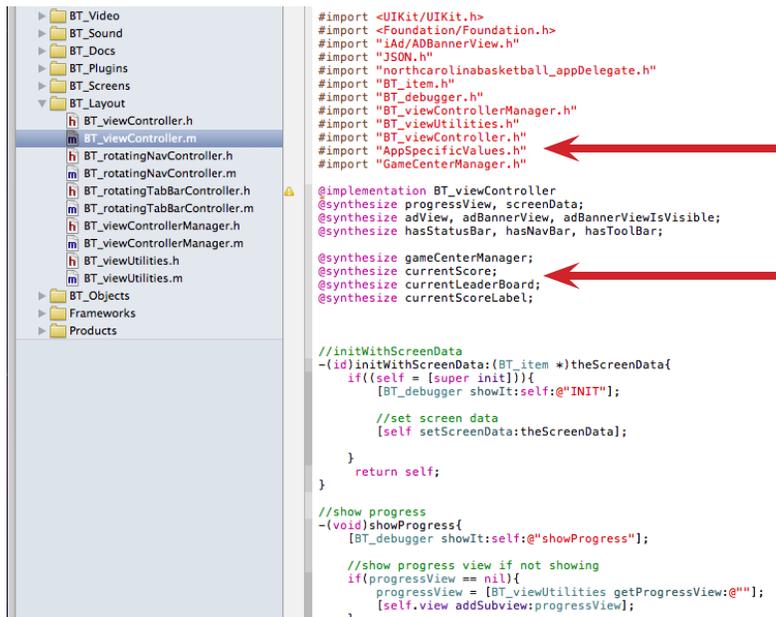
@property (nonatomic, retain) GameCenterManager *gameCenterManager;
@property (nonatomic, assign) int64_t currentScore;
@property (nonatomic, retain) NSString* currentLeaderBoard;
@property (nonatomic, retain) UILabel *currentScoreLabel;

- (IBAction) showLeaderboard;
```

5. Select the “BT_viewController.m” file and add the following code:

```
#import “AppSpecificValues.h”
#import “GameCenterManager.h”
```

```
@synthesize gameCenterManager;
@synthesize currentScore;
@synthesize currentLeaderBoard;
@synthesize currentScoreLabel;
```



```
BT_Video
BT_Sound
BT_Docs
BT_Plugins
BT_Screens
BT_Layout
BT_viewController.h
BT_viewController.m
BT_rotatingNavController.h
BT_rotatingNavController.m
BT_rotatingTabBarController.h
BT_rotatingTabBarController.m
BT_viewControllerManager.h
BT_viewControllerManager.m
BT_viewUtilities.h
BT_viewUtilities.m
BT_Objects
Frameworks
Products

#import <UIKit/UIKit.h>
#import <Foundation/Foundation.h>
#import "Ad/ADBannerView.h"
#import "JSON.h"
#import "northcarolinabasketball_appDelegate.h"
#import "BT_item.h"
#import "BT_debugger.h"
#import "BT_viewControllerManager.h"
#import "BT_viewUtilities.h"
#import "AppSpecificValues.h"
#import "GameCenterManager.h"

@implementation BT_viewController
@synthesize progressView, screenData;
@synthesize adView, adBannerView, adBannerViewIsVisible;
@synthesize hasStatusBar, hasNavBar, hasToolBar;

@synthesize gameCenterManager;
@synthesize currentScore;
@synthesize currentLeaderBoard;
@synthesize currentScoreLabel;

//initWithScreenData
-(id)initWithScreenData:(BT_item *)theScreenData{
    if([self = [super init]]){
        [BT_debugger showIt:self:@"INIT"];

        //set screen data
        [self setScreenData:theScreenData];
    }
    return self;
}

//show progress
-(void)showProgress{
    [BT_debugger showIt:self:@"showProgress"];

    //show progress view if not showing
    if(progressView == nil){
        progressView = [BT_viewUtilities getProgressView:@""];
        [self.view addSubview:progressView];
    }
}
```

Integrating Game Center

Insert the following code after the “@synthesize” statements:

```
//GameCenter View Did Load
- (void)viewDidLoad {
    [super viewDidLoad];

    self.currentLeaderBoard = kLeaderboardID;
    self.currentScore = 0;

    if ([GameCenterManager isGameCenterAvailable]) {
        self.gameCenterManager = [[[GameCenterManager alloc] init] autorelease];
        [self.gameCenterManager setDelegate:self];
        [self.gameCenterManager authenticateLocalUser];
    } else {
        // The current device does not support Game Center.
    }
}

- (IBAction) showLeaderboard
{
    GKLeaderboardViewController *leaderboardController = [[GKLeaderboardViewController alloc] init];
    if (leaderboardController != NULL)
    {
        leaderboardController.category = self.currentLeaderBoard;
        leaderboardController.timeScope = GKLeaderboardTimeScopeWeek;
        leaderboardController.leaderboardDelegate = self;
        [self presentModalViewController: leaderboardController animated: YES];
    }
}

- (void)leaderboardViewControllerDidFinish:(GKLeaderboardViewController *)viewController
{
    [self dismissModalViewControllerAnimated: YES];
    [viewController release];
}

- (IBAction) showAchievements
{
    GKAchievementViewController *achievements = [[GKAchievementViewController alloc] init];
    if (achievements != NULL)
    {
        achievements.achievementDelegate = self;
        [self presentModalViewController: achievements animated: YES];
    }
}

- (void)achievementViewControllerDidFinish:(GKAchievementViewController *)viewController;
{
    [self dismissModalViewControllerAnimated: YES];
    [viewController release];
}
```

Integrating Game Center

Add this code as well, below the code you just added:

```
//GameCenter View Did Unload
- (void)viewDidUnload
{
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;

    self.gameCenterManager = nil;
    self.currentLeaderBoard = nil;
    self.currentScoreLabel = nil;
}

- (void)dealloc {
    [gameCenterManager release];
    [currentLeaderBoard release];
    [currentScoreLabel release];
    [super dealloc];
}
```

6. Select the Localizable.strings file. It's located in the BT_Config folder.

Add the following under the “//quiz” section:

```
“quizSubmitScore” = “Submit Score”;
```

7. Select the BT_screen_quiz.h file in the BT_Screens folder.

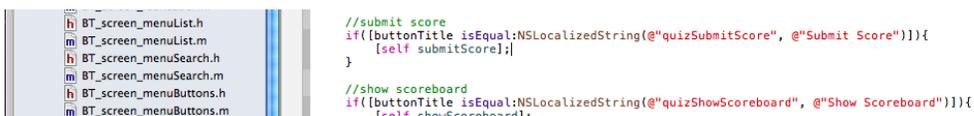
Add the following code:

```
-(void)submitScore;
```

8. Select the BT_screen_quiz.m file.

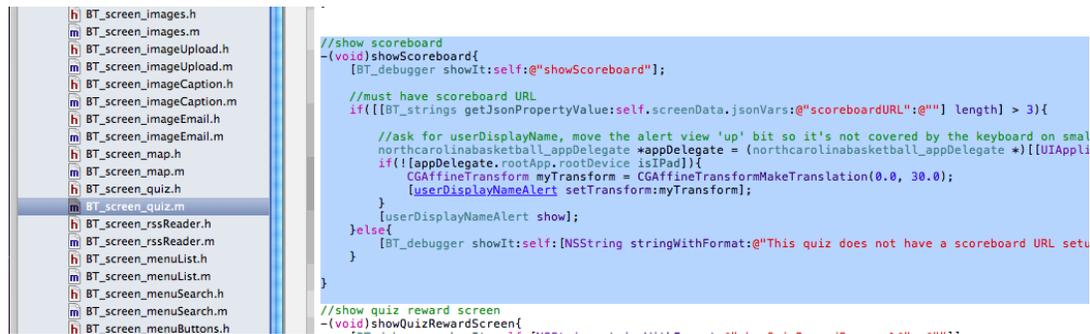
Find “//delegate method for action sheet clicks” section and add the following:

```
//submit score
if([buttonTitle isEqual:NSString(@"quizSubmitScore", @"Submit Score")]){
    [self submitScore];
}
```



Integrating Game Center

Delete the section between //show scoreboard and //show quiz reward screen



```
BT_screen_images.h
BT_screen_images.m
BT_screen_imageUpload.h
BT_screen_imageUpload.m
BT_screen_imageCaption.h
BT_screen_imageCaption.m
BT_screen_imageEmail.h
BT_screen_imageEmail.m
BT_screen_map.h
BT_screen_map.m
BT_screen_quiz.h
BT_screen_quiz.m
BT_screen_rssReader.h
BT_screen_rssReader.m
BT_screen_menuList.h
BT_screen_menuList.m
BT_screen_menuSearch.h
BT_screen_menuSearch.m
BT_screen_menuButtons.h

//show scoreboard
-(void)showScoreboard{
    [BT_debugger showIt:self:@"showScoreboard"];

    //must have scoreboard URL
    if([[BT_strings getJsonPropertyValue:self.screenData.jsonVars:@"scoreboardURL:@""] length] > 3){

        //ask for userDisplayName, move the alert view 'up' bit so it's not covered by the keyboard on small
        northcarolinabasketballAppDelegate *appDelegate = (northcarolinabasketballAppDelegate *)[UIApplication.sharedApplication.delegate];
        if(![appDelegate.rootApp.rootDevice.isiPad]){
            CGAffineTransform myTransform = CGAffineTransformMakeTranslation(0.0, 30.0);
            [userDisplayNameAlert setTransform:myTransform];
        }
        [userDisplayNameAlert show];
    }else{
        [BT_debugger showIt:self:[NSString stringWithFormat:@"This quiz does not have a scoreboard URL setu
    }
}

//show quiz reward screen
-(void)showQuizRewardScreen{
```

Above “//show quiz reward screen” Add:

```
//submit score
-(void)submitScore{
    if(totalPoints > 0)
    {
        [self.gameCenterManager reportScore: totalPoints forCategory: self.currentLeaderBoard];
    }

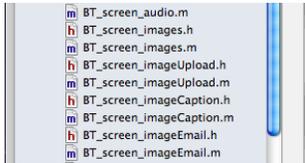
    GKLeaderboardViewController *leaderboardController = [[GKLeaderboardViewController
alloc] initWithCategory:self.currentLeaderBoard];
    if (leaderboardController != NULL)
    {
        leaderboardController.category = self.currentLeaderBoard;
        leaderboardController.timeScope = GKLeaderboardTimeScopeWeek;
        leaderboardController.leaderboardDelegate = self;
        [self presentViewController: leaderboardController animated: YES];
    }
}

//show scoreboard
-(void)showScoreboard{
    GKLeaderboardViewController *leaderboardController = [[GKLeaderboardViewController
alloc] initWithCategory:self.currentLeaderBoard];
    if (leaderboardController != NULL)
    {
        leaderboardController.category = self.currentLeaderBoard;
        leaderboardController.timeScope = GKLeaderboardTimeScopeWeek;
        leaderboardController.leaderboardDelegate = self;
        [self presentViewController: leaderboardController animated: YES];
    }
}
```

Integrating Game Center

Above “//scoreboard button shows if we have it ‘on’ and if a scoreboardURL is provided,” add:

```
//Submit Score Button
[finishedButtons addObject:NSMakeRange(@"quizSubmitScore", @"Submit Score")];
```



```
}
}
//Submit Score Button
[finishedButtons addObject:NSMakeRange(@"quizSubmitScore", @"Submit Score")];
//scoreboard button shows if we have it "on" and if a scoreboardURL is provided
if([[BT_strings getJsonPropertyValue:self.screenData.jsonVars:@"showScoreboard":@"1"] isEqualToString:@"1"]
    || ([[BT_strings getJsonPropertyValue:self.screenData.jsonVars:@"scoreboardURL":@""] length] > 0)]
    [finishedButtons addObject:NSMakeRange(@"quizShowScoreboard", @"Show Scoreboard")];
}
```

9. Select AppSpecificValues.h file.

Delete the following:

```
#define kHardLeaderboardID @"com.appledts.HardTapList"
#define kAwesomeLeaderboardID @"com.appledts.AwesomeTapList"

//Achievement IDs
#define kAchievementGotOneTap @"com.appletest.one_tap"
#define kAchievementHidden20Taps @"com.appledts.twenty_taps"
#define kAchievementBigOneHundred @"com.appledts.one_hundred_taps"
```

Change the following:

```
#define kEasyLeaderboardID @"com.appledts.EasyTapList"
```

to:

```
#define kLeaderboardID @"com.appledts.EasyTapList"
```

Finally, replace “com.appledts.EasyTapList” with your id you created in step one in iTunes Connect.

10. Run your app and test it out.

If you’ve done it right, you’ll get one of two screens when your app launches:

If you’re already signed into Game Center, you’ll see something like this:



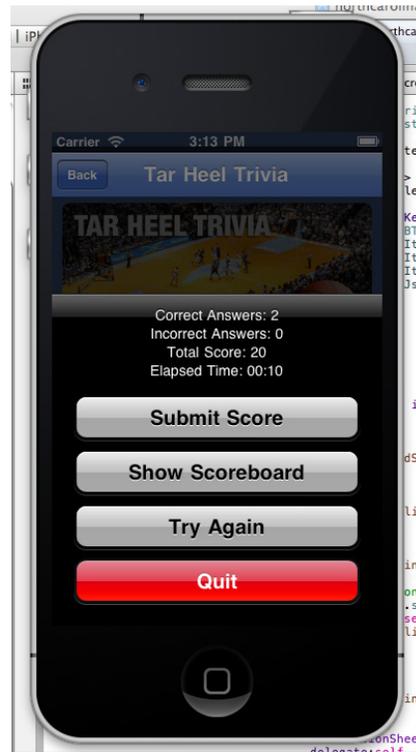
Or, you’ll be asked to sign into Game Center:



Integrating Game Center

Additional Screenshots:

This is what the popup screen should look like at the conclusion of a quiz:



When a user submits their score or presses Show Scoreboard, this is what they will see:

