

Gamekit is a framework that provides leader board, achievements, and more features to an iOS application. In this tutorial, we will be explaining the steps involved in adding a leader board and updating the score.

Steps Involved

Step 1. In iTunes connect, ensure that you have a **unique App ID** and when we create the application update with the **bundle ID** and code signing in Xcode with corresponding provisioning profile.

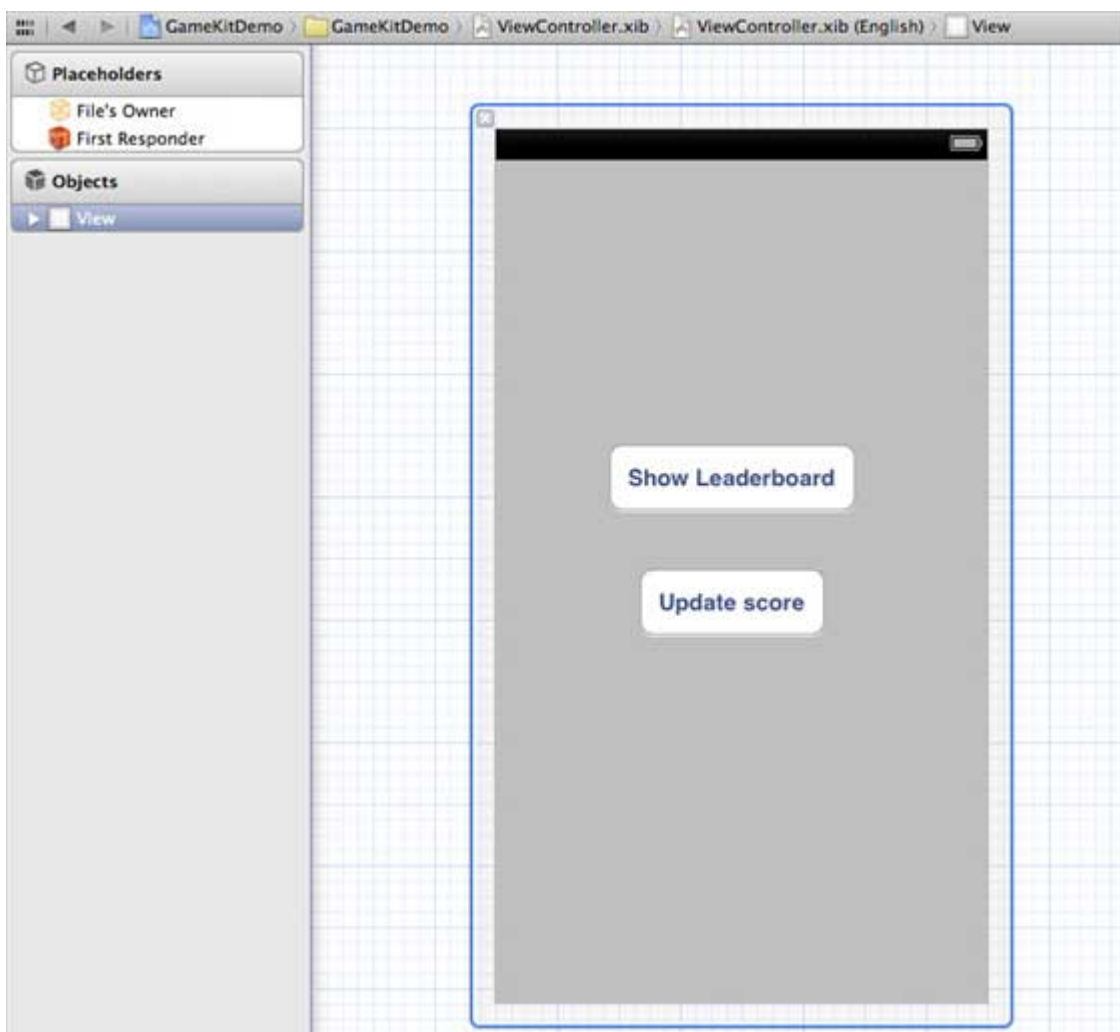
Step 2. Create a new application and update application information. You can know more about this in apple-add new apps documentation.

Step 3. Setup a leader board in **Manage Game Center** of your application's page where add a single leaderboard and give **leaderboard ID** and score Type. Here we give leader board ID as tutorialsPoint.

Step 4. The next steps are related to handling code and creating UI for our application.

Step 5. Create a **single view application** and enter the **bundle identifier** is the identifier specified in **iTunes connect**.

Step 6. Update the ViewController.xib as shown below –



Step 7. Select your project file, then select **targets** and then add **GameKit.framework**.

Step 8. Create **IBActions** for the buttons we have added.

Step 9. Update the **ViewController.h** file as follows –

```
#import <UIKit/UIKit.h>
#import <GameKit/GameKit.h>

@interface ViewController : UIViewController
<GKLeaderboardViewControllerDelegate>

- (IBAction)updateScore:(id)sender;
- (IBAction)showLeaderBoard:(id)sender;

@end
```

Step 10. Update **ViewController.m** as follows –

```
#import "ViewController.h"

@interface ViewController ()

@end

@implementation ViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
    if([[GKLocalPlayer localPlayer].authenticated == NO])
    {
        [[GKLocalPlayer localPlayer]
         authenticateWithCompletionHandler:^(NSError *error)
         {
             NSLog(@"Error%@", error);
         }]];
    }
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

- (void) updateScore: (int64_t) score
forLeaderboardID: (NSString*) category
{
    GKScore *scoreObj = [[GKScore alloc]
initWithCategory:category];
scoreObj.value = score;
scoreObj.context = 0;
[scoreObj reportScoreWithCompletionHandler:^(NSError *error) {
    // Completion code can be added here
    UIAlertView *alert = [[UIAlertView alloc]
initWithTitle:nil message:@"Score Updated Succesfully"
delegate:self cancelButtonTitle:@"Ok" otherButtonTitles: nil];
[alert show];

}]];
}

- (IBAction)updateScore:(id)sender{
    [self updateScore:200 forLeaderboardID:@"tutorialsPoint"];
}

- (IBAction)showLeaderBoard:(id)sender{
    GKLeaderboardViewController *leaderboardViewController =
[[GKLeaderboardViewController alloc] init];
leaderboardViewController.leaderboardDelegate = self;
[self presentViewController:
leaderboardViewController animated:YES];
}

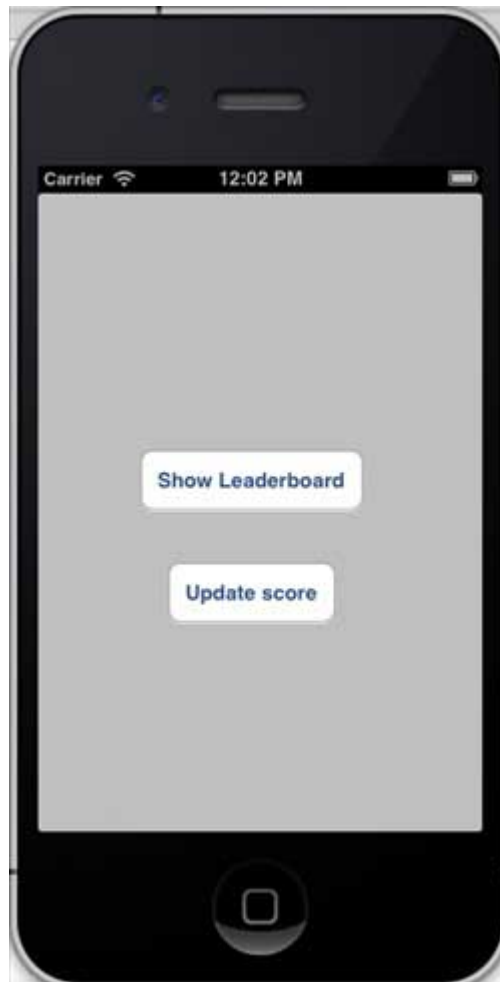
}
```

```
#pragma mark - Gamekit delegates
- (void)leaderboardViewControllerDidFinish:
(GKLeaderboardViewController *)viewController{
    [self dismissModalViewControllerAnimated:YES];
}

@end
```

Output

When we run the application, we'll get the following output –



When we click "show leader board", we would get a screen similar to the following –





When we click "update score", the score will be updated to our leader board and we will get an alert as shown below –

