

CSCI 6907 Project Final Report

iBurn: Earphone Run-in Application with Twitter Post

Yinchong Han

Dec 6th. 2012

Project Introduction

Headphone is the pathway between people and music devices. Good sound quality of a headphone not only based on its material and technology but also requires user to use it in a right way. Just like a new car, a new earphone or headphone needs a period of break in time which is called “burn-in” in professional sound editing field. Play white noise, pink noise, or some special sound in right volume for a certain period of time will help the headphone to get its perfect condition as soon as possible.

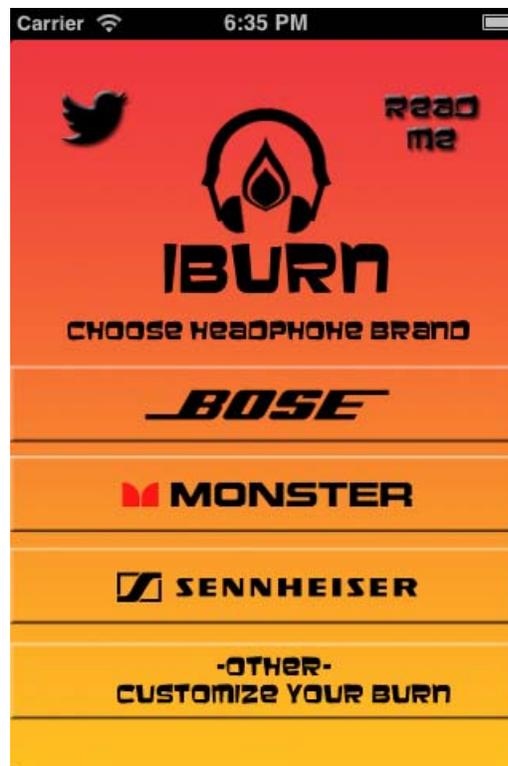


Figure 1: Main View of iBurn

iBurn is a headphone burn-in application with Twitter share post function. The app contains some professional edited audio files to help you to burn-in. Also, different brand headphones are very different, so the iBurn has special edited sound tracks for some popular brands of headphones such as Bose, Monster, and Sennheiser.

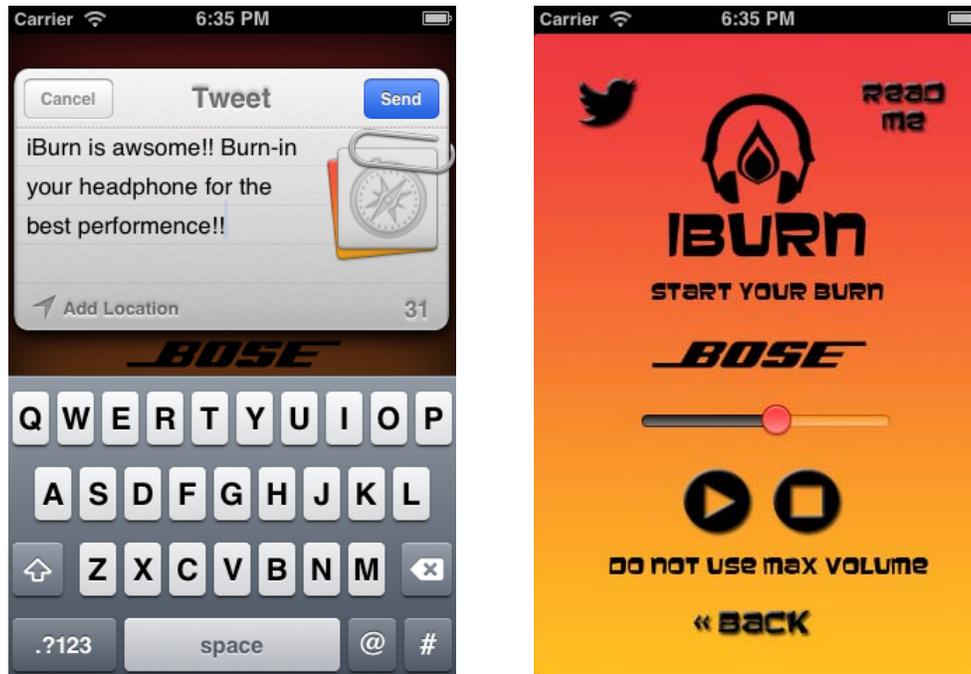


Figure 2: Twitter Post Page and Bose Burn-in Page

User will only need to choose the model of the headphone, then connect the headphone to iPhones or other iOS devices and run the app. The app will help user to burn-in the headphone during the nighttime. After a certain period of time, the headphone will reach the best condition of sound quality.

Also, the Twitter share and post function is included in the iBurn. Users can share their experience via built-in Twitter post function, or they can just simply click on the Twitter Logo button to spread the word of my app. **All code, sound files, and images (except the logos of Twitter, Bose, Monster, and Sennheiser) are original.**

Problems Encountered and Solutions

During coding this project, the most confused part is the new Xcode 4. I was very familiar with old version Xcode, however, it is really different of using Xcode4. The biggest problem I have encountered was a very simple bug, but it is incredibly hard to fix. I can easily add sound board on one ViewController, however, the play button stop working when I try to move them into other Views. I tried to add alert to the functions and I found the alert can be run,

but the play music functions were not. Thus, I tried to use AudioToolbox and MediaPlayer frameworks without AVFoundation framework. However, the result was same. I frustrated on this problem for over two days. Every soundboard I create is working, but it stops working when I add more views. Finally, iBurn functions correctly after I changed some code, modified stopSound function and changed the way of link buttons with methods.

```
- (IBAction)playFile4:(id)sender
{
    NSString *fileName = [NSString stringWithFormat:@"white1"];
    NSString *soundFilePath = [[NSBundle mainBundle] pathForResource:fileName ofType:@"wav"];
    NSURL *soundFileURL = [NSURL fileURLWithPath:soundFilePath];

    player = [[AVAudioPlayer alloc] initWithContentsOfURL:soundFileURL error:nil];
    player.volume = 0.4;
    player.numberOfLoops = 0;
    player.delegate = self;
    [player play];
    /*UIButton *button = (UIButton *)sender;

    if ([button.titleLabel.text isEqualToString:@"Play"]) {
        [button setTitle:@"Pause" forState:UIControlStateNormal];
        [player play];
    } else if ([button.titleLabel.text isEqualToString:@"Pause"]) {
        [button setTitle:@"Play" forState:UIControlStateNormal];
        [player pause];
    }*/
}

- (IBAction)changeVolume:(id)sender
{
    UISlider *slider = (UISlider *)sender;

    [player setVolume:slider.value];
}

- (IBAction)stopSound:(id)sender
{
    if(player && player.isPlaying) {
        [player stop];
    }
}
```

Figure 3: Sample Code1

The other issue that is not a problem for my iBurn is that I have to professionally edit those noise files for different brands of headphones. I contacted one of my best friends who is working at Bose Sound Lab. He taught me how to edit sound files and how to control the frequency range. I spend more than a week to edit the sound files. I think a great app should not only have beautiful UI and great functions but also needs the best quality content that is accurate and helpful. For my iBurn App, I have to consider about different headphones need different burn-in plans; they are showed in readme pages for each brand of headphones.

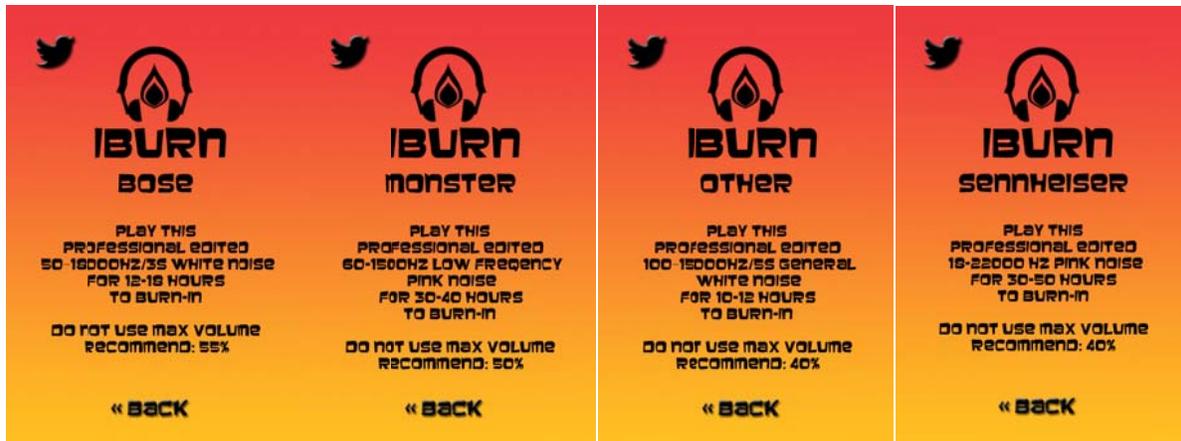


Figure3: Readme Pages

People love to use Facebook and Twitter Post in app. I tried to add both of them in my app. However, I only add Twitter successfully. Facebook have many sample apps online, which help people to understand their APIs and learn how to use them. Unfortunately, after downloading everything the samples need, I still can't run those samples. I did some research online about it, and I found that it may be caused by the new Xcode4 and new iOS system. Because I decided to run my app on the newest iOS 5 and iOS 6, I dropped the idea of embedding Facebook Post in iBurn. Use Twitter Post in app is much easier than before since iOS 5 and iOS 6 has combined Twitter. We only need to add some amount of code to active Twitter Post. Below is the sample code of embed Twitter Post in my app:

```

- (IBAction)tweetTapped:(id)sender
{
    //Check if device can send tweet
    if ([[TWTweetComposeViewController canSendTweet]])
    {
        //Create tweet sheet and set initial text
        TWTweetComposeViewController *tweetSheet =
        [[TWTweetComposeViewController alloc] init];
        [tweetSheet setInitialText:@"iBurn is awesome!! Burn-in your headphone for the best performance!!"];
        [tweetSheet addURL:[NSURL URLWithString:@"http://home.gwu.edu/~han"];
        [tweetSheet addImage:[UIImage imageNamed:@"launch.png"]];

        //Show tweet sheet
        [self presentViewController:tweetSheet animated:YES];
    }
    else
    {
        //Device cannot send Tweet. Show error notification
        UIAlertView *alertView = [[UIAlertView alloc]
        initWithTitle:@"Unable to Tweet"
        message:@"Please ensure you have at least one Twitter account setup and have internet connectivity. -iBurn"
        delegate:self
        cancelButtonTitle:@"OK"
        otherButtonTitles:nil];

        [alertView show];
    }
}

```

Figure 4: Code Sample 2: Twitter Post Function

Conclusions

This is a great experience to make a useful and complete app from scratch. I went through the entire life cycle of my application including find the requirement, make the specifications, design the architecture and layout, design and draw images, create sound files, coding, testing and debugging, and demonstration. I am planning to make minor modifications to my iBurn App and then put it into Apple App Store.