**ASSIGNMENT 01**

***Question 01:***

|  |  |
| --- | --- |
| [screenshot1](https://github.com/babygau/iosdevelopment/blob/master/resources/assignment1ss1.png) | [screenshot2](https://github.com/babygau/iosdevelopment/blob/master/resources/assignment1ss2.png) |
| [screenshot3](https://github.com/babygau/iosdevelopment/blob/master/resources/assignment1ss3.png) | [screenshot4](https://github.com/babygau/iosdevelopment/blob/master/resources/assignment1ss4.png) |

***Question 02:***

The program was crashed at its very first start due to missing #import "CrashAndBurnAppDelegate.h" and wrong parameter in main method. The last parameter should be NSStringFromClass([CrashAndBurnAppDelegate class])

After fix the bug, the program successfully loaded, but crash again appeared when clicking on button. Tracing the bug, I found the problem was at ViewDidLoad method, which assign \_nextIndex variable to an NSMutableArray which was wrong because \_nextIndex is supposed to be an NSInteger. To fix it, I assigned \_nextIndex = 0 and the program running fine. The image is switched each time the button was clicked.

**-** (**void**)**viewDidLoad**

{

[super viewDidLoad;

self.imageHistory **=** [NSMutableArray array];

*//\_nextIndex = (NSInteger)self.imageHistory;*

\_nextIndex**=**0;

}

However, I encountered another crash when kept clicking button, it turned out the \_nextIndex was greater than number of file, hence array imagePath was out of bound. Here is the fix:

**-** (NSString**\***) **\_nextImageFilename**

{

NSString **\***nextFilename;

NSArray **\***imagePaths **=** [[NSBundle mainBundle] pathsForResourcesOfType**:**@"jpg" inDirectory**:**@"Assets"];

*// Double-check to make sure the array isn't out of bounds*

**if** (\_nextIndex **<** [imagePaths count])

{

nextFilename **=** [imagePaths objectAtIndex**:**\_nextIndex];

\_nextIndex**++**;

} **else** {

\_nextIndex **=** 0;

nextFilename **=** [imagePaths objectAtIndex**:**\_nextIndex];

\_nextIndex**++**;

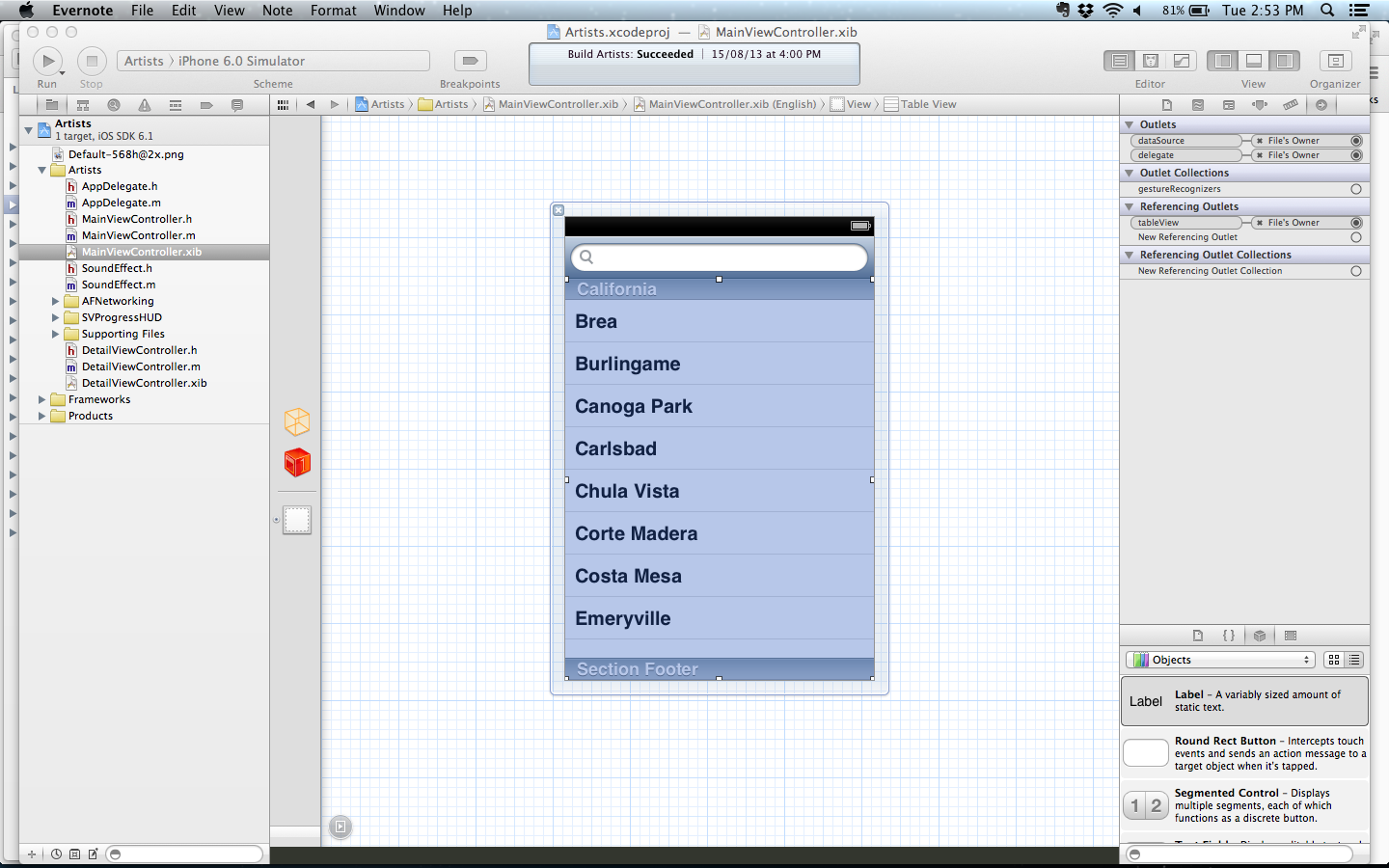
}

**return** nextFilename;

}

Generally speaking, although XCode debugging looked strange for me when I was first getting started with XCode, it provide every features that other debuggers of other programming language platform. Actually, debugging code is one of my weakness skill, I don’t usually make use of all debugging techniques (lack of knowledge) so I cannot say how much difference between XCode and others.

***Question 3:***

[](https://github.com/babygau/iosdevelopment/blob/master/resources/xib%20file.png)

xib is the extension name of nib file which stands for Nextstep Interface Builder. These days, you will usually develop your interface using a file format whose extension is .xib; when your app is built, your target’s .xib files are translated (“compiled”) into .nib format. .xib files are stored in XML based format. With nib files, you create and manipulate your user interfaces graphically, using XCode, instead of programmatically. Because you can see the results of your changes instantly, you can experiment with different layouts and configurations very quickly. You can also change many aspects of your user interface later without rewriting any code.

Xib contain the following elements

* IB Archive <archive/>
* IB Data <data/>
* IB Boolean <bool> or <Boolean/>
* IB Integer <int/> or <integer/>
* IB Float <float/> or <real/>
* IB Double <double/>
* IB String <string/>
* IB Bytes <bytes/>
* IB Nil <nil/>
* IB Reference <reference/>
* IB Object <object/>