**Unity3D实现ISO录音功能**

Posted on 2013年07月09日 by U3d / [Unity3D脚本/插件](http://www.unitymanual.com/category/script) /被围观 24 次

在工程内添加这两个framework，我这里给工程命名audio\_text，在生成的audio\_textViewController.h里的代码如下：

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 001 | view plain |
| 002 |  |
| 003 | #import |
| 004 | #import |
| 005 | #import |
| 006 |  |
| 007 | @**interface** audio\_textViewController : UIViewController { |
| 008 |  |
| 009 | IBOutlet UIButton \*bthStart; |
| 010 | IBOutlet UIButton \*bthPlay; |
| 011 | IBOutlet UITextField \*freq; |
| 012 | IBOutlet UITextField \***value**; |
| 013 | IBOutlet UIActivityIndicatorView \*actSpinner; |
| 014 | **BOOL** toggle; |
| 015 |  |
| 016 | *//Variable setup for access in the class* |
| 017 | NSURL \*recordedTmpFile; |
| 018 | AVAudioRecorder \*recorder; |
| 019 | NSError \*error; |
| 020 | } |
| 021 |  |
| 022 | @property (nonatomic,retain)IBOutlet UIActivityIndicatorView \*actSpinner; |
| 023 | @property (nonatomic,retain)IBOutlet UIButton \*bthStart; |
| 024 | @property (nonatomic,retain)IBOutlet UIButton \*bthPlay; |
| 025 |  |
| 026 | -(IBAction)start\_button\_pressed; |
| 027 | -(IBAction)play\_button\_pressed; |
| 028 | @end |
| 029 | audio\_textViewController.m |
| 030 | view plain |
| 031 |  |
| 032 | #import "audio\_textViewController.h" |
| 033 |  |
| 034 | @implementation audio\_textViewController |
| 035 |  |
| 036 |  |
| 037 | - (**void**)viewDidLoad { |
| 038 | [super viewDidLoad]; |
| 039 |  |
| 040 | *//Start the toggle in true mode.* |
| 041 | toggle = YES; |
| 042 | bthPlay.hidden = YES; |
| 043 | *//Unity3D教程手册：www.unitymanual.com* |
| 044 | *//Instanciate an instance of the AVAudioSession object.* |
| 045 | AVAudioSession \* audioSession = [AVAudioSession sharedInstance]; |
| 046 | *//Setup the audioSession for playback and record.* |
| 047 | *//We could just use record and then switch it to playback leter, but* |
| 048 | *//since we are going to do both lets set it up once.* |
| 049 | [audioSession setCategory:AVAudioSessionCategoryPlayAndRecord error: &error]; |
| 050 | *//Activate the session* |
| 051 | [audioSession setActive:YES error: &error]; |
| 052 |  |
| 053 | } |
| 054 | - (IBAction) start\_button\_pressed{ |
| 055 |  |
| 056 | **if**(toggle) |
| 057 | { |
| 058 | toggle = NO; |
| 059 | [actSpinner startAnimating]; |
| 060 | [bthStart setTitle:@"停" forState: UIControlStateNormal ]; |
| 061 | bthPlay.enabled = toggle; |
| 062 | bthPlay.hidden = !toggle; |
| 063 |  |
| 064 | *//Begin the recording session.* |
| 065 | *//Error handling removed. Please add to your own code.* |
| 066 |  |
| 067 | *//Setup the dictionary object with all the recording settings that this* |
| 068 | *//Recording sessoin will use* |
| 069 | *//Its not clear to me which of these are required and which are the bare minimum.* |
| 070 | *//This is a good resource: http://www.totodotnet.net/tag/avaudiorecorder/* |
| 071 | NSMutableDictionary\* recordSetting = [[NSMutableDictionary alloc] init]; |
| 072 |  |
| 073 | [recordSetting setValue :[NSNumber numberWithInt:kAudioFormatAppleIMA4] forKey:AVFormatIDKey]; |
| 074 |  |
| 075 | [recordSetting setValue:[NSNumber numberWithFloat:[freq.text floatValue]] forKey:AVSampleRateKey]; |
| 076 | [recordSetting setValue:[NSNumber numberWithInt: [**value**.text intValue]] forKey:AVNumberOfChannelsKey]; |
| 077 |  |
| 078 | *//Now that we have our settings we are going to instanciate an instance of our recorder instance.* |
| 079 | *//Generate a temp file for use by the recording.* |
| 080 | *//This sample was one I found online and seems to be a good choice for making a tmp file that* |
| 081 | *//will not overwrite an existing one.* |
| 082 | *//I know this is a mess of collapsed things into 1 call. I can break it out if need be.* |
| 083 | recordedTmpFile = [NSURL fileURLWithPath:[NSTemporaryDirectory() stringByAppendingPathComponent: [NSString stringWithFormat: @"%.0f.%@", [NSDate timeIntervalSinceReferenceDate] \* 1000.0, @"caf"]]]; |
| 084 | NSLog(@"Using File called: %@",recordedTmpFile); |
| 085 | *//Setup the recorder to use this file and record to it.* |
| 086 | recorder = [[ AVAudioRecorder alloc] initWithURL:recordedTmpFile settings:recordSetting error:&error]; |
| 087 | *//Use the recorder to start the recording.* |
| 088 | *//Im not sure why we set the delegate to self yet.* |
| 089 | *//Found this in antother example, but Im fuzzy on this still.* |
| 090 | [recorder setDelegate:self]; |
| 091 | *//We call this to start the recording process and initialize* |
| 092 | *//the subsstems so that when we actually say "record" it starts right away.* |
| 093 | [recorder prepareToRecord]; |
| 094 | *//Start the actual Recording* |
| 095 | [recorder record]; |
| 096 | *//There is an optional method for doing the recording for a limited time see* |
| 097 | *//[recorder recordForDuration:(NSTimeInterval) 10]* |
| 098 | *//Unity3D教程手册：www.unitymanual.com* |
| 099 | } |
| 100 | **else** |
| 101 | { |
| 102 | toggle = YES; |
| 103 | [actSpinner stopAnimating]; |
| 104 | [bthStart setTitle:@"开始录音" forState:UIControlStateNormal ]; |
| 105 | bthPlay.enabled = toggle; |
| 106 | bthPlay.hidden = !toggle; |
| 107 |  |
| 108 | NSLog(@"Using File called: %@",recordedTmpFile); |
| 109 | *//Stop the recorder.* |
| 110 | [recorder stop]; |
| 111 | } |
| 112 | } |
| 113 |  |
| 114 | - (**void**)didReceiveMemoryWarning { |
| 115 | *// Releases the view if it doesn't have a superview.* |
| 116 | [super didReceiveMemoryWarning]; |
| 117 |  |
| 118 | *// Release any cached data, images, etc that aren't in use.* |
| 119 | } |
| 120 |  |
| 121 | -(IBAction) play\_button\_pressed{ |
| 122 |  |
| 123 | *//The play button was pressed...* |
| 124 | *//Setup the AVAudioPlayer to play the file that we just recorded.* |
| 125 | AVAudioPlayer \* avPlayer = [[AVAudioPlayer alloc] initWithContentsOfURL:recordedTmpFile error:&error]; |
| 126 | [avPlayer prepareToPlay]; |
| 127 | [avPlayer play]; |
| 128 |  |
| 129 | } |
| 130 |  |
| 131 | - (**void**)viewDidUnload { |
| 132 | *// Release any retained subviews of the main view.* |
| 133 | *// e.g. self.myOutlet = nil;* |
| 134 | *//Clean up the temp file.* |
| 135 | NSFileManager \* fm = [NSFileManager defaultManager]; |
| 136 | [fm removeItemAtPath:[recordedTmpFile path] error:&error]; |
| 137 | *//Call the dealloc on the remaining objects.* |
| 138 | [recorder dealloc]; |
| 139 | recorder = nil; |
| 140 | recordedTmpFile = nil; |
| 141 | } |
| 142 |  |
| 143 |  |
| 144 | - (**void**)dealloc { |
| 145 | [super dealloc]; |
| 146 | } |