```
//
//  ViewController.m
//  TheScream
//
//  Created by T. Binkowski on 5/3/12.
//  Copyright (c) 2012 University of Chicago. All rights reserved.
//

#import "ViewController.h"
#import <QuartzCore/QuartzCore.h>
#import <AudioToolbox/AudioToolbox.h>
#import <MediaPlayer/MediaPlayer.h>

@interface ViewController ()

@end

@implementation ViewController
@synthesize currentImage;
@synthesize backgroundMusic;

/
    ********************************************************************************
    *******
 * @method      viewDidLoad
 * @abstract    <# abstract #>
 * @description <# description #>
 ********************************************************************************
     ****/
- (void)viewDidLoad
{
    [super viewDidLoad];

    // Hide the status bar
    [[UIApplication sharedApplication] setStatusBarHidden:YES withAnimation:
        NO];
}

/
    ********************************************************************************
    *******
 * @method      viewDidUnload
 * @abstract    <# abstract #>
 * @description <# description #>
 ********************************************************************************
     ****/
- (void)viewDidUnload
{
    [super viewDidUnload];
}

/
    ********************************************************************************
    *******
 * @method      viewDidAppear:
 * @abstract    <# abstract #>
 * @description <# description #>
 ********************************************************************************
     ****/
- (void)viewWillAppear:(BOOL)animated {
    [super viewWillAppear:animated];

    [self becomeFirstResponder];  // For shaking detection
    [self playBackgroundMusic];
    [self animate];
}

- (void)viewWillDisappear:(BOOL)animated
{
     [self resignFirstResponder];
}

#pragma mark - Sounds
/
    ********************************************************************************
    *******
 * @method      playBackgroundMusic
 * @abstract    <# abstract #>
 * @description <# description #>
 ********************************************************************************
     ****/
- (void)playBackgroundMusic
{
    NSError *error;
    NSString *backgroundMusicPath = [[NSBundle mainBundle] pathForResource:
        @"BackgroundMusic" ofType:@"mp3"];
    NSURL *backgroundMusicURL = [NSURL fileURLWithPath:backgroundMusicPath];

    self.backgroundMusic = [[AVAudioPlayer alloc] initWithContentsOfURL:
        backgroundMusicURL error:&error];
    [self.backgroundMusic prepareToPlay];
    [self.backgroundMusic play];
}

/
    ********************************************************************************
    *******
 * @method          SoundEffects
 * @abstract        <# Abstract #>
 * @description     <# Description #>
 ********************************************************************************
     ***/
- (void)soundEffects
{
    NSString *squishPath = [[NSBundle mainBundle] pathForResource:
        @"Suspense" ofType:@"caf"];
    NSURL *squishURL = [NSURL fileURLWithPath:squishPath];
    SystemSoundID soundID;
    AudioServicesCreateSystemSoundID((__bridge CFURLRef)squishURL, &soundID)
        ;
    AudioServicesAddSystemSoundCompletion(soundID, NULL, NULL,
        MyAudioServicesSystemSoundCompletionProc, NULL);
```

```
    AudioServicesPlaySystemSound(soundID);
}

/
    ********************************************************************************
    *******
 * @method          <# Method Name #>
 * @abstract        Need to release the sound object
 * @description     <# Description #>
 ********************************************************************************
     ***/
void MyAudioServicesSystemSoundCompletionProc(SystemSoundID ssID,  void *
    clientData)
{
    NSLog(@"%s :: Release Sound", __PRETTY_FUNCTION__);
    AudioServicesDisposeSystemSoundID(ssID);
}


#pragma mark - Shake
/
    ********************************************************************************
    *******
 * @method      canBecomeFirstResponder
 * @abstract    <# abstract #>
 * @description <# description #>
 ********************************************************************************
     ****/
- (BOOL)canBecomeFirstResponder
{
    return YES;
}

- (void)motionBegan:(UIEventSubtype)motion withEvent:(UIEvent *)event {
    //if (motion != UIEventSubtypeMotionShake) return;
}

- (void)motionEnded:(UIEventSubtype)motion withEvent:(UIEvent *)event {
    if (motion == UIEventTypeMotion && event.type ==
        UIEventSubtypeMotionShake) {
        NSLog(@"%@ motionEnded", [NSDate date]);

        // Get the background view (tag==100) and remove all subviews
        UIView *background = [self.view viewWithTag:100];
        for (UIView *subview in [background subviews]) {
            [subview removeFromSuperview];
        }
    }

    if ([super respondsToSelector:@selector(motionEnded:withEvent:)]) {
        [super motionEnded:motion withEvent:event];
    }
}
- (void)motionCancelled:(UIEventSubtype)motion withEvent:(UIEvent *)event {
}
```

```
#pragma mark - Gestures
/
    ********************************************************************************
    *******
 * @method          addStar
 * @abstract        <# Abstract #>
 * @description     <# Description #>
 ********************************************************************************
     ***/
- (IBAction)addStar:(UIGestureRecognizer*)gestureRecognizer {
    NSLog(@"Add Star");

    UIView *background = gestureRecognizer.view;
    CGPoint locationInView = [gestureRecognizer locationInView:[background
        superview]];
    //NSLog(@"Tap %5.2f %5.2f",locationInView.x,locationInView.y);

    UIImageView *image;
    if (self.currentImage == nil) {
        image = [[UIImageView alloc] initWithImage:[UIImage imageNamed:
            @"star"]];
    } else {
        image = [[UIImageView alloc] initWithImage:self.currentImage];
    }

    image.transform = CGAffineTransformScale(image.transform, 0.3, 0.3);
    image.center = locationInView;
    image.userInteractionEnabled = YES;
    [self addGestureRecognizersToStar:image];

    [background addSubview:image];
    [self soundEffects];
}

/
    ********************************************************************************
    *******
 * @method          addGestureToStar:
 * @abstract        <# Abstract #>
 * @description     <# Description #>
 ********************************************************************************
     ***/
- (void)addGestureRecognizersToStar:(UIView *)piece
{
    UIRotationGestureRecognizer *rotationGesture =
        [[UIRotationGestureRecognizer alloc] initWithTarget:self action:
        @selector(rotatePiece:)];
    [piece addGestureRecognizer:rotationGesture];

    UIPinchGestureRecognizer *pinchGesture = [[UIPinchGestureRecognizer
        alloc] initWithTarget:self action:@selector(scalePiece:)];
    [pinchGesture setDelegate:self];
    [piece addGestureRecognizer:pinchGesture];

    UIPanGestureRecognizer *panGesture = [[UIPanGestureRecognizer alloc]
```

```
        initWithTarget:self action:@selector(panPiece:)];
    [panGesture setMaximumNumberOfTouches:2];
    [panGesture setDelegate:self];
    [piece addGestureRecognizer:panGesture];
}

/
    ********************************************************************
    *******
 * @method      adjustAnchorPointForGestureRecognizer
 * @abstract    <# abstract #>
 * @description scale and rotation transforms are applied relative to the
     layer's anchor point
 *              this method moves a gesture recognizer's view's anchor point
     between the user's fingers
 ********************************************************************
     ****/
- (void)adjustAnchorPointForGestureRecognizer:(UIGestureRecognizer *)
    gestureRecognizer
{
    if (gestureRecognizer.state == UIGestureRecognizerStateBegan) {
        UIView *piece = gestureRecognizer.view;
        CGPoint locationInView = [gestureRecognizer locationInView:piece];
        CGPoint locationInSuperview = [gestureRecognizer locationInView:
            piece.superview];

        piece.layer.anchorPoint = CGPointMake(locationInView.x / piece.
            bounds.size.width, locationInView.y / piece.bounds.size.height);
        piece.center = locationInSuperview;
    }
}

/
    ********************************************************************
    *******
 * @method      panPiece:
 * @abstract    <# abstract #>
 * @description shift the piece's center by the pan amount
 *              reset the gesture recognizer's translation to {0, 0} after
     applying so the next
 *              callback is a delta from the current position
 ********************************************************************
     ****/

- (void)panPiece:(UIPanGestureRecognizer *)gestureRecognizer
{
    UIView *piece = [gestureRecognizer view];
    [[piece superview] bringSubviewToFront:piece];

    [self adjustAnchorPointForGestureRecognizer:gestureRecognizer];

    if ([gestureRecognizer state] == UIGestureRecognizerStateBegan ||
        [gestureRecognizer state] == UIGestureRecognizerStateChanged) {
        CGPoint translation = [gestureRecognizer translationInView:[piece
            superview]];
        [piece setCenter:CGPointMake([piece center].x + translation.x,
            [piece center].y + translation.y)];
        [gestureRecognizer setTranslation:CGPointZero inView:[piece
            superview]];
    }
}

/
    ********************************************************************
    *******
 * @method      rotatePiece:
 * @abstract    <# abstract #>
 * @description rotate the piece by the current rotation
 *              reset the gesture recognizer's rotation to 0 after applying
     so
 *              the next callback is a delta from the current rotation
 ********************************************************************
     ****/
- (void)rotatePiece:(UIRotationGestureRecognizer *)gestureRecognizer
{
    [self adjustAnchorPointForGestureRecognizer:gestureRecognizer];

    if ([gestureRecognizer state] == UIGestureRecognizerStateBegan ||
        [gestureRecognizer state] == UIGestureRecognizerStateChanged) {
        [gestureRecognizer view].transform = CGAffineTransformRotate
            ([[gestureRecognizer view] transform], [gestureRecognizer
            rotation]);
        [gestureRecognizer setRotation:0];
    }
}

/
    ********************************************************************
    *******
 * @method      scalePiece
 * @abstract
 * @description Scale the piece by the current scale; reset the gesture
     recognizer's
 *              rotation to 0 after applying so the next callback is a delta
     from the current scale
 ********************************************************************
     ****/
- (void)scalePiece:(UIPinchGestureRecognizer *)gestureRecognizer
{
    [self adjustAnchorPointForGestureRecognizer:gestureRecognizer];

    if ([gestureRecognizer state] == UIGestureRecognizerStateBegan ||
        [gestureRecognizer state] == UIGestureRecognizerStateChanged) {
        [gestureRecognizer view].transform = CGAffineTransformScale
            ([[gestureRecognizer view] transform], [gestureRecognizer scale]
            , [gestureRecognizer scale]);
        [gestureRecognizer setScale:1];
    }
}

#pragma mark - Button Target Actions
```

```
/
    ********************************************************************************
    *******
 * @method          photoButton
 * @abstract        <# Abstract #>
 * @description     <# Description #>
 ********************************************************************************
     ***/
- (IBAction)photoButton:(id)sender
{
    UIImagePickerController *imagePicker = [[UIImagePickerController alloc]
        init];

    // If our device has a camera, we want to take a picture, otherwise, we
    // just pick from photo library
    if ([UIImagePickerController isSourceTypeAvailable:
        UIImagePickerControllerSourceTypeCamera]) {
        [imagePicker setSourceType:UIImagePickerControllerSourceTypeCamera];
    } else {
        [imagePicker setSourceType:
            UIImagePickerControllerSourceTypePhotoLibrary];
    }

    // This line of code will generate 2 warnings right now, ignore them
    imagePicker.delegate = self;

    // Show image picker on the screen
    [self presentModalViewController:imagePicker animated:YES];
}

/
    ********************************************************************************
    *******
 * @method      showInstructions
 * @abstract    <# abstract #>
 * @description <# description #>
 ********************************************************************************
     ****/
- (IBAction)showInstructions:(id)sender
{
    UIActionSheet *msg = [[UIActionSheet alloc]
                          initWithTitle:@"1. Tap the screen to add stars.\n"
                          "2. Move or resize the stars by dragging and
                              pinching.\n"
                          "3. Select a new image to add by clicking the
                              camera.\n"
                          "4. Shake to start over.\n"
                          delegate:nil
                          cancelButtonTitle:nil  destructiveButtonTitle:nil
                          otherButtonTitles:@"Okay", nil];
    [msg showInView:self.view];
}

- (IBAction)showAlert:(id)sender
{
    UIAlertView *av = [[UIAlertView alloc] initWithTitle:@"Title"
                                                 message:@"Hello"
                                                delegate:self
                                       cancelButtonTitle:@"OK"
                                       otherButtonTitles:@"A",@"B",nil];
    [av show];

}

#pragma mark - Photo Delegate
/
    ********************************************************************************
    *******
 * @method      imagePickerController:
 * @abstract    <# abstract #>
 * @description <# description #>
 ********************************************************************************
     ****/
- (void)imagePickerController:(UIImagePickerController *)picker
    didFinishPickingMediaWithInfo:(NSDictionary *)info
{
    // Get picked image from info dictionary
    UIImage *image = [info objectForKey:UIImagePickerControllerOriginalImage
        ];
    self.currentImage = image;

    // Take image picker off the screen - you must call this dismiss method
    [self dismissModalViewControllerAnimated:YES];
}

#pragma mark - Animation Effects
/
    ********************************************************************************
    *******
 * @method          <# Method Name #>
 * @abstract        <# Abstract #>
 * @description     <# Description #>
 ********************************************************************************
     ***/
- (void)animate
{
    [self soundEffects];

    CGRect offscreen = CGRectMake(0, 500, 200, 344);

    UIImageView *cat = [[UIImageView alloc] initWithImage:[UIImage
        imageNamed:@"cat"]];
    cat.frame = offscreen;
    [self.view addSubview:cat];

    [UIView animateWithDuration:4.0 delay:0.5 options:
        UIViewAnimationOptionCurveEaseInOut
                     animations:^{
                         cat.center = self.view.center;
                     }
                     completion:^(BOOL  completed){
                         // Nested animation block
```

```
                        NSLog(@"Shocked cat arrives");
                        [UIView animateWithDuration:1.0 delay:1.0 options:
                            UIViewAnimationCurveEaseOut
                                        animations:^{
                                          cat.transform =
                                              CGAffineTransformScale(cat.
                                              transform, 20, 20);
                                        }
                                        completion:^(BOOL  completed){
                                            NSLog(@"Shocked cat leaves.");
                                            [cat removeFromSuperview];
                                        }
                         ];
                    }
     ];
}

#pragma mark - Alerts View Delegate
/
    ********************************************************************
    *******
 * @method          <# Method Name #>
 * @abstract        <# Abstract #>
 * @description     <# Description #>
 ********************************************************************
     ***/
- (void)alertView:(UIAlertView *)alertView clickedButtonAtIndex:(NSInteger)
    buttonIndex
{
    printf("User selected button %d\n",buttonIndex);
}

@end
```