A01-演示和思路

2014年10月15日 星期三　　0:33

练习演示

思路：

1、显示界面，实现大图功能
2、下一题
3、判断答案是否正确
4、提示

A02-创建项目

2014年10月15日 星期三　　0:34

1、创建项目
2、拖入图片

A03-设置图标和启动屏

2014年10月15日 星期三　　10:54

1、把图片放在Images.xcassets，Images.xcassets只有iOS7+支持，iOS6前不支持
　　Images.xcassets中的图片编译后被打包进Assets.car这个文件
2、如果程序程序部署的时候设置为6.0，图片还是会放在bundle的根文件夹下

retina屏幕（高清屏幕）
30*30pt（point）的图片　在3.5inch非retina屏幕就是30*30px
　　在3.5inch retina屏幕就是60*60px
3.5 inch　320*480pt　px

3.5 inch retina 640*960px （320*480pt）

4.0 inch retina 640*1136px（320*568pt）

4.7 inch retina 750*1334px（375*667pt）

5.5 inch retina 1080*1920px （414*736pt）

A04-摆控件

2014年10月15日 星期三　　10:54

放置控件的时候讲解控件的属性

1、放置程序上半部分的控件

控件的拖拽的先后顺序会影响显示的先后位置

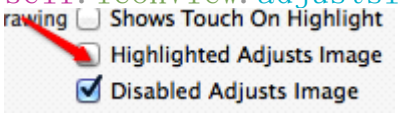
设置完控件后

要设置coin不能被点击

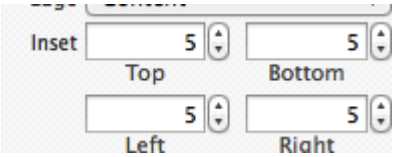
```
self.coinView.userInteractionEnabled = NO;
```

去掉图片按钮的高亮显示属性（还可以用代码设置）

```
self.iconView.adjustsImageWhenHighlighted = NO;
```



设置图片按钮的白边（按钮的内边距）

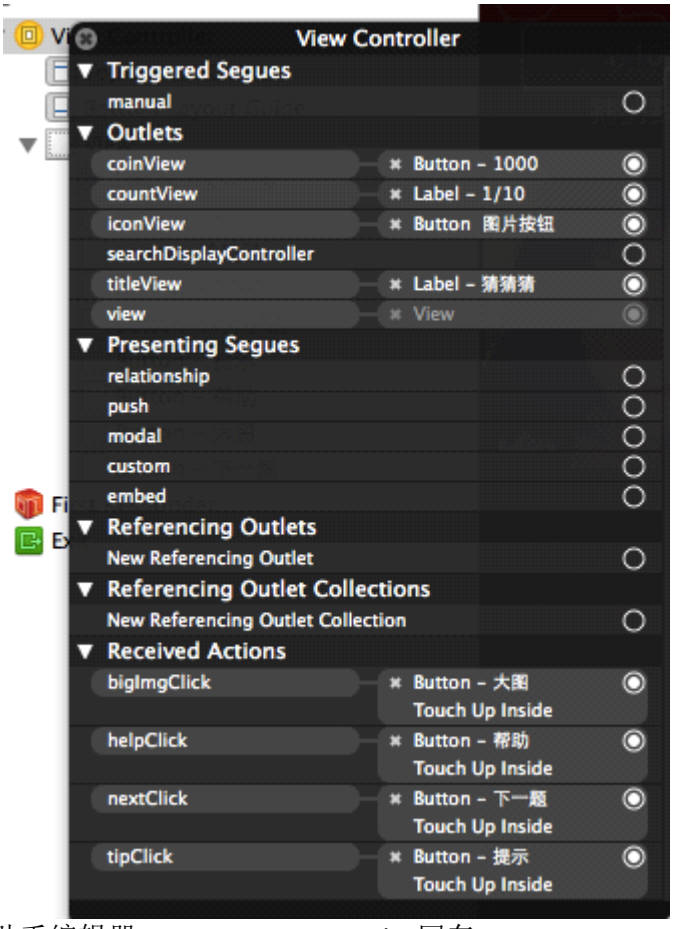


2、隐藏或高亮状态栏

```
- (BOOL)prefersStatusBarHidden
{
    return YES;
}
- (UIStatusBarStyle)preferredStatusBarStyle
{
    return UIStatusBarStyleLightContent;
}
```

3、连线需要操作的控件

注意：在连线的时候，如果连错线，注意一定要删除连线，否则会出错



打开助手编辑器 option + command + 回车
关闭助手编辑器 command + 回车

A05-点击放大和缩小图片

2014年11月18日 星期二　　09:36

1、点击大图按钮，放大图片

```
//1 点击大图按钮，放大图片
- (IBAction)bigImageClick {
    //1.1 放大图片(改变frame)
    CGFloat iconW = self.view.frame.size.width;
    CGFloat iconH = iconW;
    CGFloat iconX = 0;
    CGFloat iconY = (self.view.frame.size.height - iconW) / 2;
    self.iconView.frame = CGRectMake(iconX, iconY, iconW, iconH);
}
```

2、点击放大图片，先要把自动布局去掉

3、添加动画效果

```
//1.2 添加动画效果
[UIView animateWithDuration:1.0 animations:^{
    self.iconView.frame = CGRectMake(iconX, iconY, iconW, iconH);
}];
```

4、添加遮盖view

```
//1.3 添加遮盖view
UIButton *coverView = [[UIButton alloc] init];
[self.view addSubview:coverView];
//设置遮盖view的大小和父view一样
coverView.frame = self.view.bounds;
coverView.backgroundColor = [UIColor blackColor];
//设置半透明
coverView.alpha = 0.5;

//1.4 把图片设置在屏幕最前面
[self.view bringSubviewToFront:self.iconView];
```

5、给遮盖view添加动画效果

```
//设置半透明 （先设置完全透明，再慢慢改变成半透明（动画效果））
coverView.alpha = 0;

//1.5 添加动画效果 (代码的顺序自己调整)
[UIView animateWithDuration:1.0 animations:^{
    self.iconView.frame = CGRectMake(iconX, iconY, iconW, iconH);
```

```
        coverView.alpha = 0.5;
    }];
```

6、点击遮盖层还原图片大小

```
    //1.5 给遮盖view注册事件
    [coverView addTarget:self action:@selector(smallImgClick)
forControlEvents:UIControlEventTouchUpInside];
    self.coverView = coverView;

    //1.6 点击遮盖view，还原小图
    - (void)smallImgClick
    {
        [UIView animateWithDuration:1.0 animations:^{
            //动画的方式还原小图和移除遮盖层
            self.iconView.frame = CGRectMake(88, 133, 200, 200);
            self.coverView.alpha = 0;
        } completion:^(BOOL finished) {
            [self.coverView removeFromSuperview];
            self.coverView = nil;
        }];
    }
```

7、点击iconView放大或缩小图片

```
    //1.7 点击iconView放大或缩小图片
    - (IBAction)iconViewClick {
        //判断当前图片是放大还是原来状态
        if (self.coverView == nil) {
            //如果遮盖层不存在，说明当前是小图状态
            [self bigImageClick];
        }else{
            [self smallImgClick];
        }
    }
```

A06-加载plist，字典转模型

2014年10月16日 星期四　　0:32

//写模型初始化方法的组合键

command + 左　光标到当前行最后

shift + option + 上　选中要复制的行

command + c

control + command + 上/下　切到对应的.m文件

command + v

1、controller中加载plist和字典转模型

```
//定义存储模型的数组
@property (nonatomic, strong) NSArray *questions;



//2 从plist中懒加载数据
- (NSArray *)questions
{
    if (_questions == nil) {
        NSBundle *bundle = [NSBundle mainBundle];
        //2.1 文件路径
        NSString *path = [bundle pathForResource:@"questions" ofType:@"plist"];
        //2.2 加载数据
        NSArray *dicArray = [NSArray arrayWithContentsOfFile:path];
        //2.3 遍历字典转换成模型，添加到临时数组中
        NSMutableArray *tmpArray = [NSMutableArray array];
        for (NSDictionary *dic in dicArray) {
            CZQuestion *question = [[CZQuestion alloc] init];
            question.answer = dic[@"answer"];
            question.icon = dic[@"icon"];
            question.title = dic[@"title"];
            question.options = dic[@"options"];

            [tmpArray addObject:question];
        }
        //2.4
        _questions = tmpArray;
    }
    return _questions;
}
```

2、进一步封装，把初始化模型的代码封装到CZQuestion类中

```
- (instancetype)initWithDic:(NSDictionary *)dic
{
    if (self = [super init]) {
```

```
        self.answer = dic[@"answer"];
        self.icon = dic[@"icon"];
        self.title = dic[@"title"];
        self.options = dic[@"options"];
    }
    return self;
}

+ (instancetype)questionWithDic:(NSDictionary *)dic
{
    return [[self alloc] initWithDic:dic];
}

+ (NSArray *)questionsList
{
    NSBundle *bundle = [NSBundle mainBundle];
    //2.1 文件路径
    NSString *path = [bundle pathForResource:@"questions" ofType:@"plist"];
    //2.2 加载数据
    NSArray *dicArray = [NSArray arrayWithContentsOfFile:path];
    //2.3 遍历字典转换成模型，添加到临时数组中
    NSMutableArray *tmpArray = [NSMutableArray array];
    for (NSDictionary *dic in dicArray) {
        CZQuestion *question = [[CZQuestion alloc] init];
        question.answer = dic[@"answer"];
        question.icon = dic[@"icon"];
        question.title = dic[@"title"];
        question.options = dic[@"options"];

        [tmpArray addObject:question];
    }
    return tmpArray;
}
```

3、改造controller中的懒加载 如果此处不理解，设置断点调试

```
/*
 * 文档注释
 * 2 从plist中懒加载数据
 */

- (NSArray *)questions
{
    if (_questions == nil) {
        _questions = [CZQuestion questionsList];
    }
    return _questions;
}
```

4、介绍代码库的使用

<# name #>

A07-下一题

2014年10月15日 星期三　　10:54

1、controller中添加一个属性，记录当前题目的索引

```
/*
 * 记录当前题目的索引
 */
@property (nonatomic, assign) int index;
```

2、点击下一题按钮

```
/*
 * 3 下一题 加载下一题的数据
 */
- (IBAction)nextClick {
    self.index ++;
    //3.1 取得当前题目
    CZQuestion *question = self.questions[self.index];

    //3.2 给控件赋值
    self.titleView.text = question.title;
    [self.iconView setImage:[UIImage imageNamed:question.icon] forState:UIControlStateNormal];
    self.countView.text = [NSString stringWithFormat:@"%d/%lu",self.index + 1,self.questions.count];

    //3.3 判断当前是否是最后一张图片，如果是最后一张，禁用下一题按钮
//    if (self.index == self.questions.count - 1) {
//        self.nextView.enabled = NO;
//    }
    self.nextView.enabled = self.index != self.questions.count - 1;
}
```

3、view加载完，显示第一题

```
//在viewDidLoad中
self.index--;
[self nextClick];
```

4、动态生成选项的按钮

5、动态生成答案的按钮

A08-动态生成答案按钮

2014年10月15日 星期三　　10:54

1、在controller中添加两个属性

```
/*
 * 放置答案按钮的容器
 */
@property (weak, nonatomic) IBOutlet UIView *answersView;
/*
 * 放置选项按钮的容器
 */
@property (weak, nonatomic) IBOutlet UIView *optionsView;
```

2、生成输入答案所需的按钮并添加到answersView上

```
//4 生成输入答案所需的按钮并添加到answersView上
- (void)setAnswerButtons:(CZQuestion *)question
{

    //删除之前的按钮
//    for (UIButton *btn in self.answersView.subviews) {
//        [btn removeFromSuperview];
//    }

    [self.answersView.subviews makeObjectsPerformSelector:@selector(removeFromSuperview)];


    NSUInteger answerCount = question.answer.length;
    CGFloat margin = 10;
    CGFloat btnW = 35;
    CGFloat btnH = 35;
    //计算第一个按钮左边的距离
    CGFloat leftMargin = (self.answersView.frame.size.width - answerCount * btnW - (answerCount - 1) * margin) / 2;
    CGFloat btnY = 0;

    for (int i = 0; i < answerCount; i++) {
        UIButton *answerBtn = [UIButton buttonWithType:UIButtonTypeCustom];
        [self.answersView addSubview:answerBtn];
        CGFloat btnX = leftMargin + i * (btnW + margin);

        answerBtn.frame = CGRectMake(btnX, btnY, btnW, btnH);
        //设置按钮的背景图片
        [answerBtn setBackgroundImage:[UIImage imageNamed:@"btn_answer"] forState:UIControlStateNormal];
        [answerBtn setBackgroundImage:[UIImage imageNamed:@"btn_answer_highlighted"]
forState:UIControlStateHighlighted];
        //设置按钮文字的颜色
        [answerBtn setTitleColor:[UIColor blackColor] forState:UIControlStateNormal];
    }

}
```

A09-动态生成选项按钮

2014年10月15日 星期三　　10:54

```
//5 生成输入选项所需的按钮并添加到optionsView上
- (void)setOptionButtons:(CZQuestion *)question
{
    //移除上一次的按钮
    [self.optionsView.subviews makeObjectsPerformSelector:@selector(removeFromSuperview)];

    CGFloat btnW = 35;
    CGFloat btnH = 35;
    int columnCount = 7;
    CGFloat marginX = (self.optionsView.frame.size.width - columnCount * btnW) / (columnCount + 1);
    CGFloat marginY = 10;

    for (int i = 0; i < question.options.count; i++) {
        UIButton *btn = [UIButton buttonWithType:UIButtonTypeCustom];
        [self.optionsView addSubview:btn];

        int row = i / columnCount;
        int column = i % columnCount;

        CGFloat btnX = marginX + column * (btnW + marginX);
        CGFloat btnY = row * (btnH + marginY);

        btn.frame = CGRectMake(btnX, btnY, btnW, btnH);
        //设置按钮的背景图片
        [btn setBackgroundImage:[UIImage imageNamed:@"btn_option"] forState:UIControlStateNormal];
        [btn setBackgroundImage:[UIImage imageNamed:@"btn_option_highlighted"]
forState:UIControlStateHighlighted];
        //设置按钮的文字颜色和选项文字
        [btn setTitleColor:[UIColor blackColor] forState:UIControlStateNormal];
        [btn setTitle:question.options[i] forState:UIControlStateNormal];
    }
}
```

A10-处理选项按钮的点击

2014年10月15日 星期三　　10:54

1、点击选项按钮找到第一个空白 的答案按钮，并把选项的内容赋值，隐藏当前选项按钮

1.1在生成选项按钮的时候给选项按钮注册事件

```
//给选项按钮注册事件
[btn addTarget:self action:@selector(setOptionToAnswer:) forControlEvents:UIControlEventTouchUpInside];
```

1.2 实现

```
/*
 *  6 点击选项按钮
 */
- (void)setOptionToAnswer:(UIButton *)sender
{
    //6.1 隐藏当前点击的选项按钮
    sender.hidden = YES;
    //6.2 点击选项按钮设置 答案
    //找到第一个空白的答案按钮，设置文字
    for (UIButton *answerBtn in self.answersView.subviews) {

        //answerBtn.currentTitle 获取当前按钮上的文字  只读
        //判断按钮文字是否为空
        if (answerBtn.currentTitle == nil) {
            [answerBtn setTitle:sender.currentTitle forState:UIControlStateNormal];
            break;
        }
    }
}
```

A11-处理答案按钮的点击

2014年10月15日 星期三　　10:54

1、在生成答案按钮的方法中，注册事件

```
//注册事件
[answerBtn addTarget:self action:@selector(setAnswerToOption:) forControlEvents:UIControlEventTouchUpInside];
```

2、点击答案按钮，找到 对应的选项按钮，显示它，并清空当前点击的按钮

```
/*
 * 7 点击答案按钮
 */
- (void)setAnswerToOption:(UIButton *)sender
{
    //7.1 还原选项按钮
    for (UIButton *optionBtn in self.optionsView.subviews) {
        //判断当前按钮和选项按钮中的文字是否一样
        if ([sender.currentTitle isEqualToString:optionBtn.currentTitle]) {
            optionBtn.hidden = NO;
            break;
        }
    }
    //7.2 清空当前点击的按钮
    [sender setTitle:nil forState:UIControlStateNormal];
}
```

3、如果选项按钮中有重复的文字 这个时候会出问题。解决方案，使用tag来表示选项按钮

3.1 在//5 生成输入选项所需的按钮并添加到optionsView上 中添加

```
//标示
btn.tag = i;
```

3.2 点击选项按钮的时候，让答案按钮记录tag

```
answerBtn.tag = sender.tag;
```

3.3 在点击答案按钮中判断tag

```
if (sender.tag == optionBtn.tag) {
    optionBtn.hidden = NO;
    break;
}
```

A12-判断答案是否正确

2014年10月15日 星期三 10:54

1、点击选项按钮判断用户是否输入完答案,并记录用户输入的答案
2、如果输入完成判断答案是否正确
3、如果答案正确，文字设置成蓝色
4、如果答案错误，文字设置成红色
5、点击答案按钮，把文字设置成黑色
6、如果答案正确加分
7、解决之前的问题：答案输入完成后不允许再点击选项按钮

1、点击选项按钮判断用户是否输入完答案,并记录用户输入的答案

```
    //6.3 判断答案是否正确
    //6.3.1 判断答案是否输入完成 并记录当前输入的按钮
    BOOL isFull = YES;  //假设答案输入完成
    NSMutableString *inputAnswer = [NSMutableString string];
    for (UIButton *answerBtn in self.answersView.subviews) {
        if (answerBtn.currentTitle == nil) {
            isFull = NO;
            break;
        }
        [inputAnswer appendString:answerBtn.currentTitle];
    }
```

2、如果输入完成判断答案是否正确
3、如果答案正确，文字设置成蓝色
4、如果答案错误，文字设置成红色

```
    //6.3.2 答案输入完成判断答案是否正确
    //获取题目信息
    CZQuestion *question = self.questions[self.index];
    if (isFull) {
        if ([question.answer isEqualToString:inputAnswer]) {
            //如果答案正确 答案按钮的字体设置为蓝色
            [self setAnswerBtnColor:[UIColor blueColor]];
        }else{
            //如果答案红色 答案按钮的字体设置为红色
            [self setAnswerBtnColor:[UIColor redColor]];
        }
    }



    /*
     * 设置答案按钮的颜色
     */
    - (void)setAnswerBtnColor:(UIColor *)color
    {
        for (UIButton *answerBtn in self.answersView.subviews) {
            [answerBtn setTitleColor:color forState:UIControlStateNormal];
        }
    }
```

5、点击答案按钮，把文字设置成黑色

```
//6.3.3 点击答案按钮。把文字设置成黑色
[self setAnswerBtnColor:[UIColor blackColor]];
```

6、如果答案正确加分

```
//6.3.4 如果答案正确加分
int coin = [self.coinView.currentTitle intValue];
coin += 500;
[self.coinView setTitle:[NSString stringWithFormat:@"%d",coin] forState:UIControlStateNormal];
```

7、解决之前的问题：答案输入完成后不允许再点击选项按钮

7.1

```
// 答案输入完成禁止继续点击选项按钮
self.optionsView.userInteractionEnabled = NO;
```

7.2 点击答案按钮。允许继续点击选项按钮

```
self.optionsView.userInteractionEnabled = YES;
```

A13-答对自动进入下一题

2014年10月15日 星期三　　10:54

1、答对自动进入下一题

```
//答对自动进入下一题  延时
[self performSelector:@selector(nextClick) withObject:nil afterDelay:1.0];
```

2、进入下一题后，选项按钮不能点击（因为答对之后禁用了）

在生成选项按钮的方法中允许选项按钮点击

```
//5 生成输入选项所需的按钮并添加到optionsView上
- (void)setOptionButtons:(CZQuestion *)question
{
    //允许选项按钮点击
    self.optionsView.userInteractionEnabled = YES;
```

3、如果已经是最后一题，继续答对，程序崩溃

因为，调用nextClick的时候没用判断是否是最后一题

```
/*
 * 3 下一题 加载下一题的数据
 */
- (IBAction)nextClick {
    self.index ++;


    //判断是否是最后一题
    if (self.index == self.questions.count) {
        //提示过关
        UIAlertView *view = [[UIAlertView alloc] initWithTitle:@"提示"
            message:@"恭喜过关" delegate:nil cancelButtonTitle:@"取消"
            otherButtonTitles:@"确定", nil];
        [view show];

//        UIActionSheet *sheet = [[UIActionSheet alloc] initWithTitle:@"提
    示" delegate:nil cancelButtonTitle:@"取消" destructiveButtonTitle:@"确定"
    otherButtonTitles:nil, nil];
//        [sheet showInView:self.view];
        return;
    }
```

A14-kvc

2014年10月16日 星期四　　0:40

key value coding

```
        //self.answer = dic[@"answer"];
//        self.icon = dic[@"icon"];
//        self.title = dic[@"title"];
//        self.options = dic[@"options"];
        //kvc 把字典中的值，赋给当前对象制定的属性（@"answer"）
//        [self setValue:dic[@"answer"] forKeyPath:@"answer"];
        //遍历字典中所有的key，并把和对象属性对应的key 赋值
        [self setValuesForKeysWithDictionary:dic];
```

```
    JSPerson *person = [[JSPerson alloc] init];
    //kvc设置值
    [person setValue:@"zs" forKeyPath:@"name"];
    [person setValue:@18 forKeyPath:@"age"];

    //kvc取值
    [person valueForKeyPath:@"name"];
    [[person valueForKeyPath:@"age"] intValue];

    //把对象转成字典，（可能要存储到plist中）
    NSDictionary *dic = [person dictionaryWithValuesForKeys:
@[@"name",@"age"]];
```

```
   JSPerson *p1 = [[JSPerson alloc] init];
    p1.name = @"zs";
    p1.age = 18;
```

```
    JSPerson *p2 = [[JSPerson alloc] init];
    p2.name = @"ls";
    p2.age = 19;

    JSPerson *p3 = [[JSPerson alloc] init];
    p3.name = @"ww";
    p3.age = 16;

    NSArray *persons = @[p1,p2,p3];

    //取出数组中所有人的名字
    //方法1  循环 取出

    //方法2
    NSArray *names = [persons valueForKeyPath:@"name"];
```

```
JSPerson *p1 = [[JSPerson alloc] init];
    p1.name = @"zs";
    p1.age = 18;
    JSBook *book = [[JSBook alloc] init];
    book.name = @"金瓶梅";
    p1.book= book;

    //根据person获取book的名字
    //1
    [p1.book valueForKeyPath:@"name"];

    //2
    //NSString *name =  p1.book.name;

    //3 区分大小写
   NSString *name = [p1 valueForKeyPath:@"book.name"];
```

A15-提示按钮

2014年10月15日 星期三 10:54

```
/*
 * 8 点击提示按钮
 */
- (IBAction)tipClick {
    //重复上一次的nextQuestion
    //还原刚进入该题的状态
    self.index--;
    [self nextClick];

    //设置第一个正确答案
    CZQuestion *question = self.questions[self.index];
    //获取答案中的第一个字符
    NSString *first = [NSString stringWithFormat:@"%C",[question.answer characterAtIndex:0]];
    for (UIButton *btn in self.optionsView.subviews) {
        NSString *option = btn.currentTitle;
        if ([first isEqualToString:option]) {
            [self setOptionToAnswer:btn];
        }
    }


    //设置分数
    int coin = [self.coinView.currentTitle intValue];
    coin -= 1000;
    [self.coinView setTitle:[NSString stringWithFormat:@"%d",coin] forState:UIControlStateNormal];

}
```

X01-掌握

2014年10月15日 星期三　　10:34

- 按钮的多功能使用
- @2x的含义
- 应用程序图标、启动图片的添加
- kvc

X02-其它

2014年10月15日 星期三　　10:34

3.5英寸屏幕，指的是屏幕对角线是3.5英寸长

X03-容易忘记的代码

2014年10月15日 星期三　　21:19

```
    //把某控件置于view的顶层
     [self.view bringSubviewToFront:self.imageView];


//调用数组中每一个对象的方法
    [self.optionsView.subviews makeObjectsPerformSelector:
@selector(removeFromSuperview)];



UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"提示"
message:@"恭喜你过关了" delegate:nil cancelButtonTitle:@"取消"
otherButtonTitles:@"确定", nil];
//        [alert show];

        UIActionSheet *sheet = [[UIActionSheet alloc]
initWithTitle:@"提示" delegate:nil cancelButtonTitle:@"取消"
destructiveButtonTitle:@"确定" otherButtonTitles:@"你猜",nil];
        [sheet showInView:self.view];
```

X04-上课笔记