

# IOS智能应用开发lab2

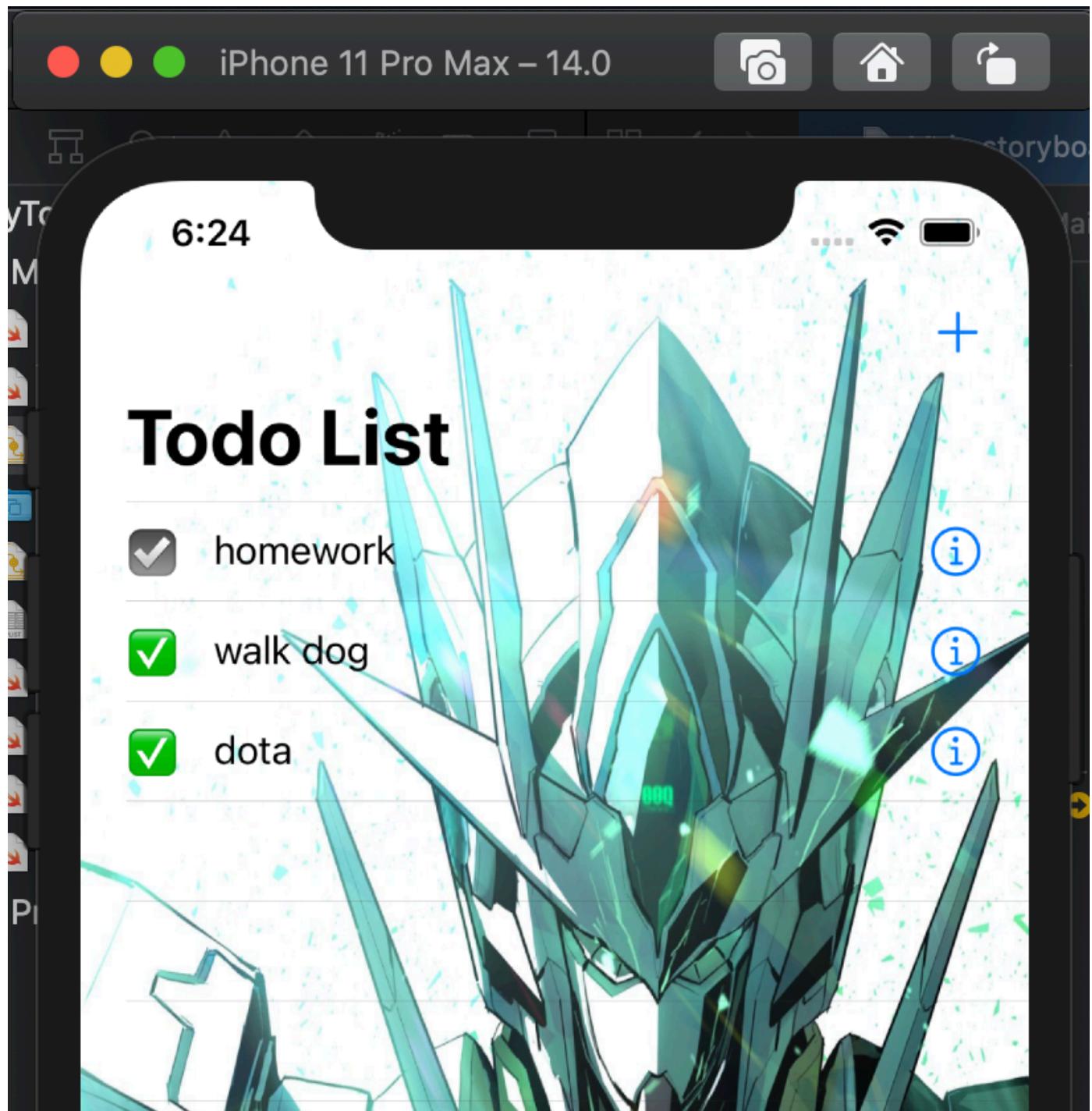
202220013徐简

## 实验结果：

跟随视频完成“代办事项”App开发，完成以下的任务

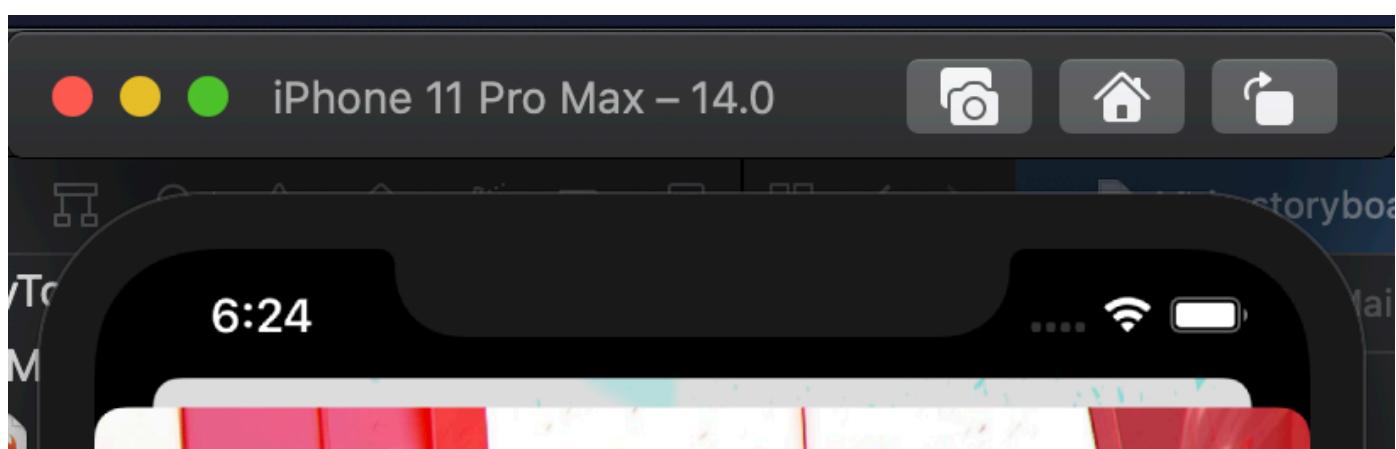
1. 实现待办事项添加并保存
2. 实现待办事项更新
3. 支持以表格中单元格右划的方式完成某个代办事项删除
4. 对界面进行个性化/美化设计。

效果如下：





Filter      ⏱ + -      Auto      ⓘ



*Cancel*

*Done*

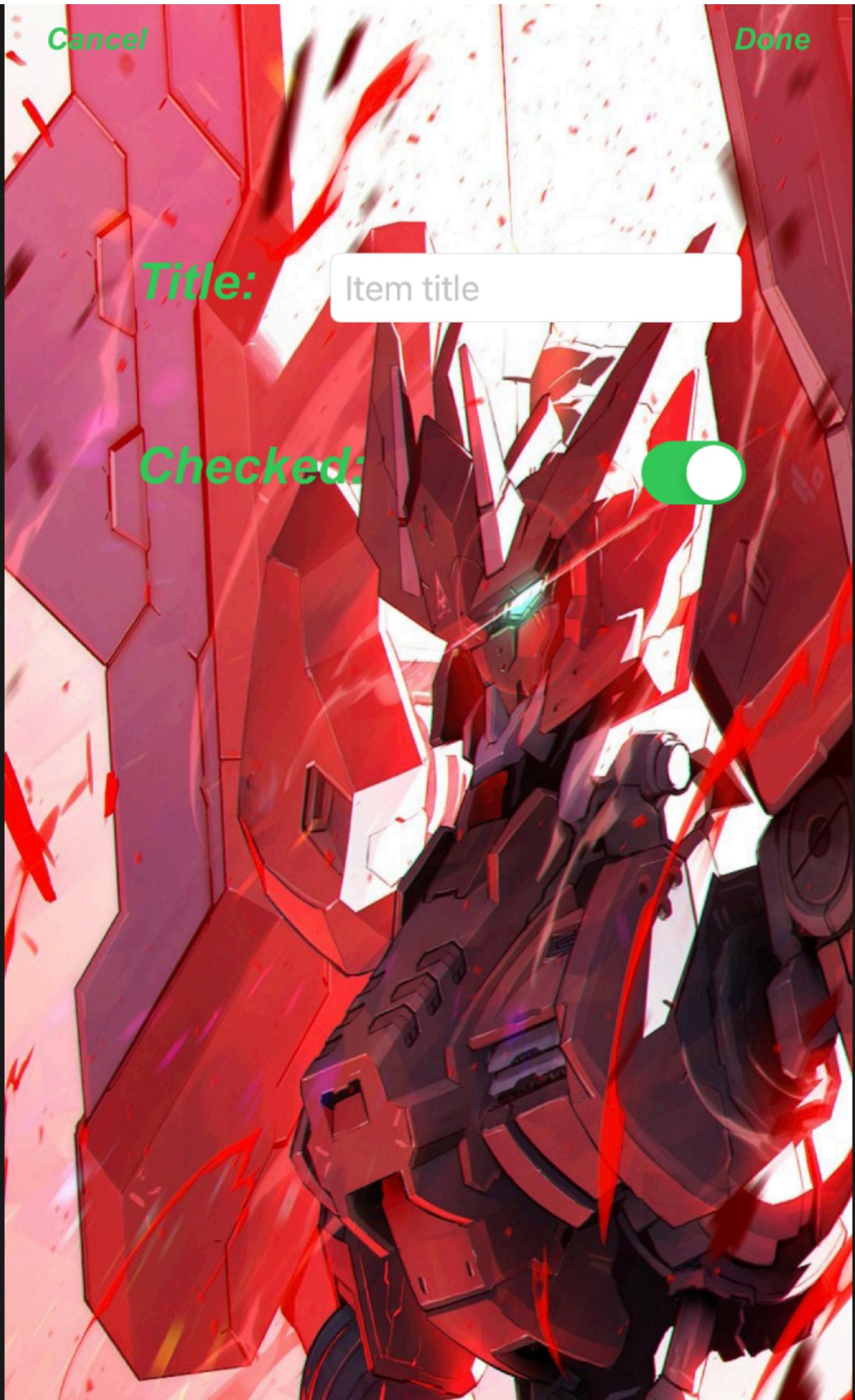
*Title:*

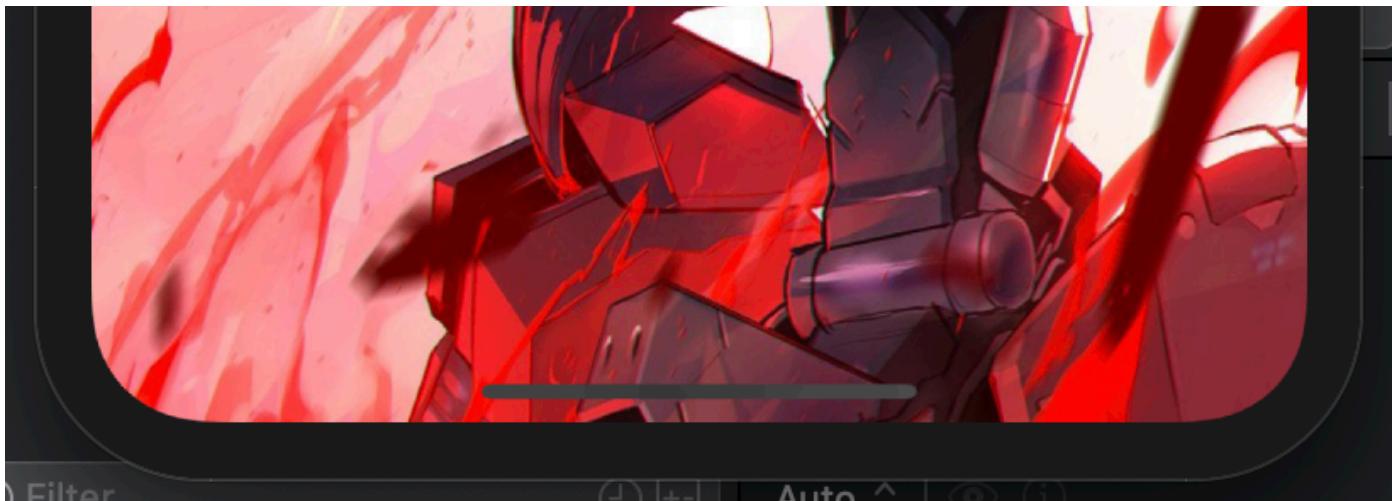
Item title

*Checked:*



P





## 代码结构

- 参考老师的实现视频，采用 `table view` 和定制 `table cell` 为主要的结构。
- 在实现编辑和修改cell内容的时候采用委托的设计方式。
- 为了实现本地保存的功能，在退出view的时候调用 `saveData`，将table中的信息保存为json格式，并在重新显示的时候再从内存中读取json数据。

## UI设计

本次实验修改了 `table view` 和 `UI view` 的背景图片设置，以及部分文本的字体样式设置。

- `table view` 的背景图片设置方法非常简单，只需要在加载时设置背景图片即可。

```
override func viewDidLoad() {
    super.viewDidLoad()
    self.navigationController?.navigationBar.prefersLargeTitles=true
    ////
    self.tableView.backgroundView = UIImageView(image: UIImage(named:
"background.png"))
    ////
    loadItems()
}
```

- 普通 `UIview` 的背景设置方法较为复杂，最初尝试用 `UIImage` 去初始化 `UIColor`，并设置背景颜色为 `UIColor`，但是发现图片无法显示完整。于是，采用堆叠 `view` 的方法来实现，将以图片生成的 `view` 放置在最下面。

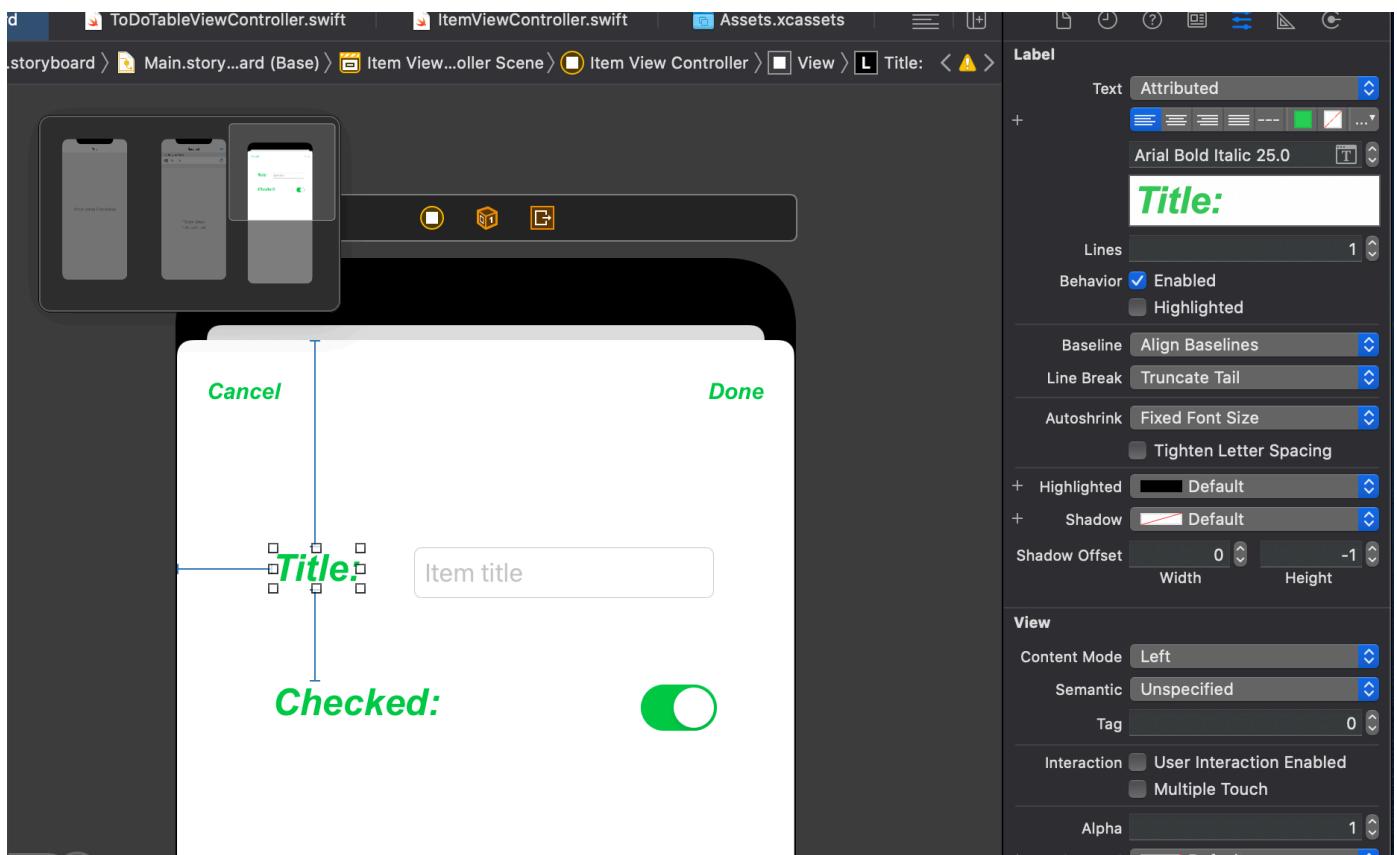
```
let image:UIImage! = UIImage(named: "itembackground.png")
self.view.backgroundColor = UIColor(patternImage: image)
```

```

let screenSize: CGRect = UIScreen.main.bounds
let screenWidth = Int(screenSize.width)
let screenHeight = Int(screenSize.height)
let uiImage = UIImage(named: "itembackground")
let imageFrame: CGRect = CGRect(x: 0, y: 0, width:(screenWidth),
height:Int(screenHeight))
let imageView = UIImageView(frame: imageFrame)
imageView.contentMode = UIView.ContentMode.scaleToFill
imageView.image = uiImage
self.view.addSubview(imageView)
self.view.sendSubviewToBack(imageView)

```

- 在字体的样式设置时，只需要在storyboard中，将字体设置为attributed，并设置相应的样式即可。



## 功能展示

见录屏

## 感想与体会

- document>>Google>>Baidu，在开发过程中，应该先研究官方文档，再去搜索引擎查，网络上关于IOS开发的内容，往往都不符合版本。
- swift的optional value与常见的语言不太一样，了解其enum实现原理后，才能? 和! 的用法掌握更好。
- 从老师精良的示例视频中，学到了很多，包括cell结构的定制、委托设计的使用等等。