

# INDIVIDUAL REPORT

FULL NAME: 薛旭  
STUDENT ID: 1612904

## DIRECTIONS

To design and develop a game of “三国华容道” based on mobile application development technologies.

诸葛亮智算华容



关云长义释曹操



## EXECUTIVE SUMMARY

REPORT NAME: 手绘风华容道 FOR IOS

### Features summary

INSTRUCTIONS: Briefly summarize the main features and the unique/special things about your game.

1. 选择关卡（卡片滑动选择）
2. 通关记录以及游戏存档
3. 重置关卡
4. 卡通画风（随机生成 Sketchy Art）

### Solution summary

INSTRUCTIONS: Briefly summarize the tools and technologies used in your game.

1. Xcode 10.2
2. Swift 5.1
3. CocoaPods (a dependency manager for Swift)
4. Cocoa UIKit
5. iPhone Xs Simulator, iOS 12

## SCREENSHOTS AND SCENARIOS

Scene 1	Scene 2	Scene 3
		
<b>Description:</b> iOS 上个 App 图标	<b>Description:</b> 选关页面，点击进入关卡，滑动切换关卡。	<b>Description:</b> 选关界面，已通关的关卡则会变为笑脸
Scene 4	Scene 5	Scene 6
		
<b>Description:</b> 游戏界面，手绘风的滑块风格为自动生成，打开关卡后可以继续上次的步骤接着玩。	<b>Description:</b> 下方两个按钮 一个为回退到选关 一个为重制关卡	<b>Description:</b> 关于界面 说明 App 的性质与分类以及制作的目的

# TECHNOLOGIES SPECIFICATION

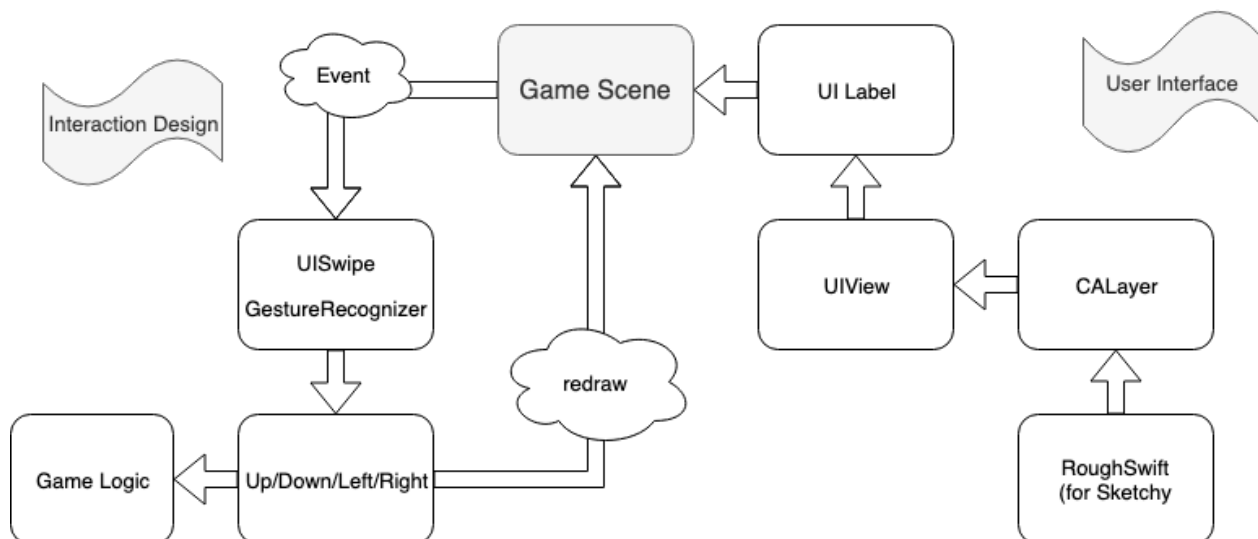
## Frontend Design and Implementation

INSTRUCTIONS: Provide additional detail about the **frontend** design and implementation in your game which should include:

- UI (User Interface)
- UE (User Experience)
- ID (Interaction Design)

本程序采取 **MVC** 架构，所以前端部分的设计归于 **View** 部分以及 **Controller** 的连接部分。

前端部分分为两个设计：一个是游戏界面，一个是选关（目录）界面，首先来看游戏界面的前端设计架构。



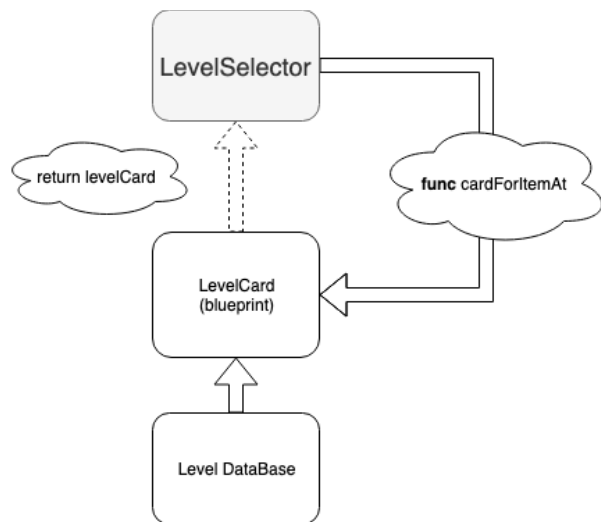
本项目基于 iOS 的 **Cocoa UIKit** 开发

架构图的左边部分是处理交互（**Interaction Design**），右边部分处理用户界面（**User Interface**）。

用户界面（手绘风）的设计采用了开源的 **RoughtSwift**，这个开源 UI 的原理是在 **CALayer** 的层面绘制图形，考虑到 **CALayer** 属于 **UIView**，每个 **UI Component** 都有一个 **UIView**，为了保持简洁，我们用 **UI Label** 来作为我们滑块的 UI 表示，通过替换 **UILabel** 的 **CALayer** 层来获得视觉效果。

交互的设计是通过通过 **Swift 4** 加入的 **UI Swipe Gesture Recognizer** 滑动手势检测器来处理，我们对于每个滑块（**UI Label**）都加入一个手势检测器，检测到滑动事件后，通过 **delegate** 机制传给我们 **ViewController** 里自定义的移动滑块函数，在 **Model** 层处理完逻辑后，通过调用自定义的 **redraw** 函数，重新绘制 **GameView** 并加入动画效果。

接下来是选关界面的前端设计：为了能让用户会心一笑，保持手绘风的卡通风格，我们同样用了 **CocoaPods** 里面一个开源项目 **VerticalCardSwiper**，每个关卡的显示类似于一个 **Table** 里面的 **Cell**，下面我们用架构图来说明

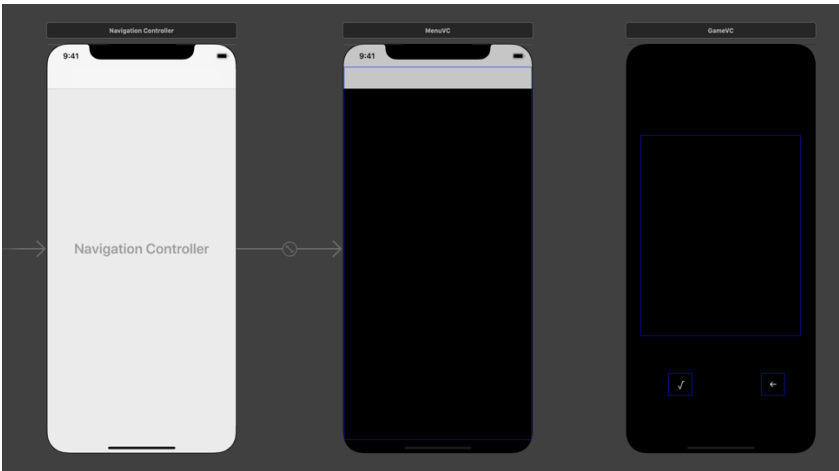


**LevelSelector** 选关器就是我们最核心的一个对象，**LevelCard** 则是我们自己绘制的蓝图。

**LevelSelector** 负责处理用户交互，用户通过点击、滑动等来选择切换关卡，**LevelSelector** 会通过 **cardForItemAt** 的协议传递给 **LevelCard**，**LevelCard** 收到信息后通过获取关卡的信息（玩家的进度等）来返回一个渲染好的对象进行显示。

显然可知 **LevelCard** 则是处理 **UI** 的，这个部分我们通过 **XCode** 自带的 **Storyboard** 来绘制。

上面所讲解的都是代码处理的部分，下面放上一张设计蓝图。  
从左到右为导航器、目录界面、游戏界面。



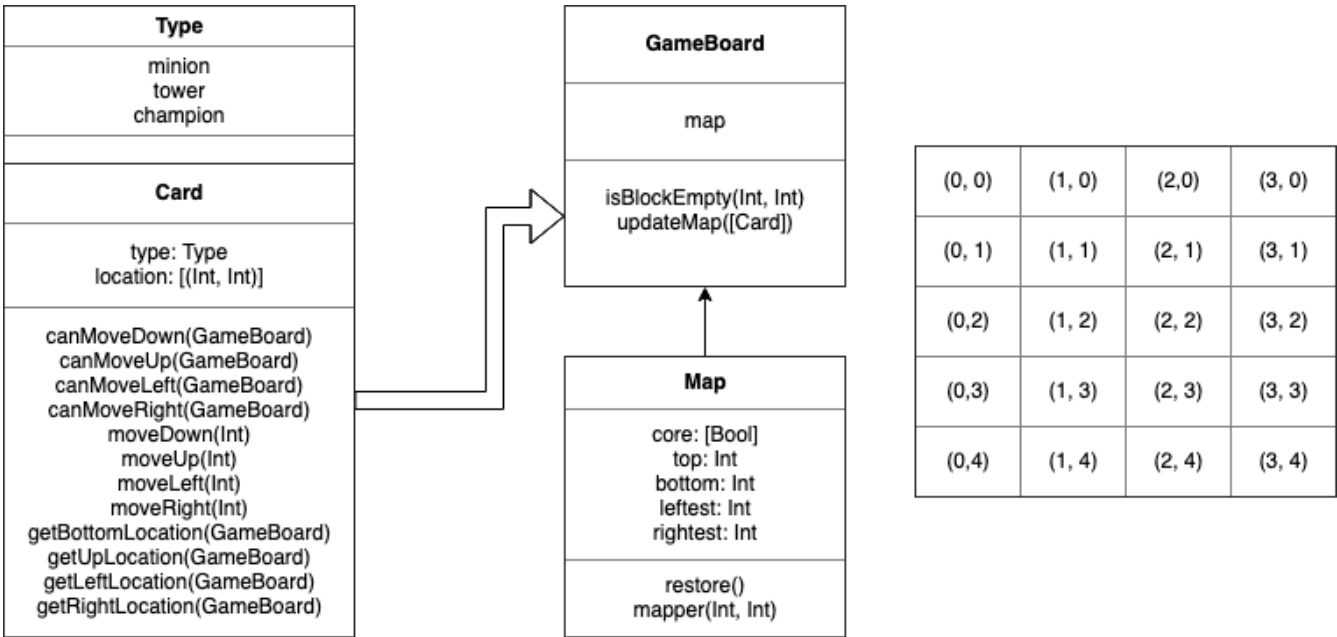
## Backend Design and Implementation

INSTRUCTIONS: Provide additional detail about the **backend** design and implementation in your game which should include:

- Game strategy
- Service architecture
- Logical algorithm
- Data storage

后端部分为了保持清晰，则根据要求的四条逐个说明。

### Game strategy



**Card** 类代表每一个滑块，考虑到滑块共有三种：2x2、2x1、1x1，在 **Card** 中加入 **type** 属性与之对应：champion、tower、minion。**Card** 的 **location** 数据是一个存着(x, y)的数组，比如曹操的位置[(1,0), (2,0), (1,1), (2,1)]，这个数据能够提供给 **ViewController** 绘制图形。

移动函数有 4 个，很简单，更新 **location** 即可。

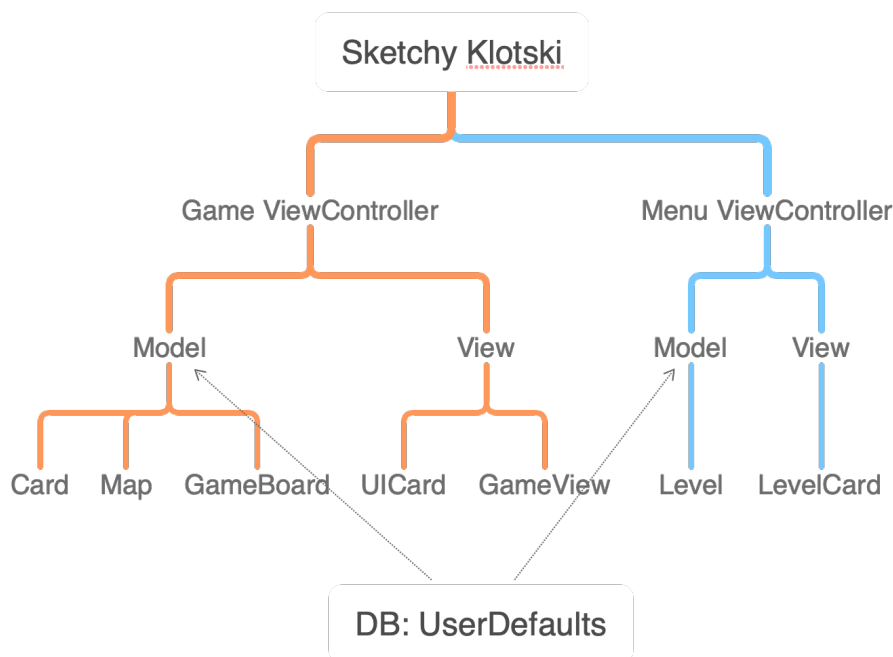
判断碰撞有 4 个函数，传入 **gameBoard**，读取 **gameBoard** 中的 **map** 实例，来判断是否能够有碰撞产生。

**Map** 类代表整个棋盘，里面存着 5\*4 个 **Bool** 数据，用来确定(x, y)位置是否为空，从而处理碰撞。

**GameBoard** 主控制整个棋盘，每当滑块移动的时候，它能够更新 **map**，以及提供接口给 **Card** 判断是否能够移动。

## Service architecture

考虑到本次项目完全参照 **MVC** 架构开设计，此处放 **MVC** 架构图



## Logical algorithm

核心算法应该在如何处理碰撞以及数据表示

可以参考在 **Game strategy** 部分的类说明，因为 **Map** 的设计让碰撞处理变得十分简单，比如一个 **Card** 实例试图往左移，首先执行 **canMoveLeft** 函数，函数里执行 **getLeftLocation** 获得 **Card** 最左边的 **x** 值，然后由接入的 **map** 判断，在棋盘上，滑块左边是否有阻挡即可，判断为 **true** 后执行 **moveDown** 函数，然后执行 **updateMap** 函数更新棋盘。

## Data storage

本项目在数据存储上，并没有直接采用 **iOS** 比较著名的 **Core Data** 框架，而是用一种比较简单的 **Key-Value** 形式存储（即 **Swift** 中封装好的 **UserDefaults**）。

当然这带来麻烦是，无法对 **Custom Class** 进行直接存储，所以我们这里采用手动 **Serialize**，将用户数据转化为 **[Int]** 来存储，就比如上文举例的曹操 **Card**，可以 **Serialize** 成 **[1,0, 2,0, 1,1, 2,1]**，我们用 **LevelSetup** 这个 **singleton Class** 来对数据进行 **encode** 和 **decode**。

## OPERATIONS SPECIFICATION

---

**INSTRUCTIONS:** Describe any important software, hardware, or other information technology that you have used to operate your game. That might include a register solution for managing your users, an achievement system to attract attention, a notification and feedback service for optimizing your game, a payment service for profit, and so on.

本项目的 **CocoaPods** 依赖有

**RoughSwift:** 给 app 加入手绘风 UI

**VerticalCardSwiper:** 给选关界面加入交互体验

**SwiftyUserDefaults:** 给数据存储提供简便性，管理用户数据。

项目架构为 **MVC**，重度使用的设计模式为 **delegate** 模式，用来处理不同组件传递信息。

目录提供了 **about** 页面，提供了联系方式与项目 **feedback** 入口。

# EXTENSIONS

---

## Game Distribution

INSTRUCTIONS: The game is supposed to be an installable and usable mobile app including necessary interaction with its backend service. Provide the URLs about your game:

- The source code URL: it may be github.com link, coding.net link, or other available link.

<https://github.com/xuuexu/SketchyKlotski>

- The released app URL: it may be Android online store link, Apple online store link, or other available link.

- The game video (<1 minute) URL: it may be .mp4, .mov, or other available video format.

YouTube: <https://youtu.be/gadpLhX4CbQ>

Bilibili: <https://www.bilibili.com/video/av55286641>

## Conclusion and Expectations

INSTRUCTIONS: Summarize your project and give future expectations about yourself and this course.

首先感谢师文轩老师开设的安卓课，让我掌握了移动端编程的基本知识，到后来选择 iOS 开发华容道后也能自我察觉到许多异同点，无论是 layout 设计中的 Constraints 机制，还是后面的 intent 以及 Navigation Controller，对本次项目的开发帮助了许多。

其次是感谢斯坦福的 CS193P 的 IOS 编程课，给了本次项目的架构以及 Swift 的使用细节很多知识来源，无论是 MVC 设计模式、CocoaUI 的绘图机制、还是处理 Touch Event，以及后面的 ViewController 的 Lifecycle 和比较难搞定的 Core Data。

本次项目算是 Swift 语言的学习项目，也是一个自我增长的项目，老项目新项目加起来 30+ commits。

项目先是能够绘图，然后能够处理手势，后来设计游戏逻辑，接着又加入了 RoughSwift UI 来变得更加美观，加入选关机制，到最后加入考虑再三，放弃了 Core Data 而直接用简单的 UserDefaults 来实现数据 Persistent。

可惜的是由于没有苹果 Developer 账号，并没有能够在 Apple Store 上架，后面希望能完善此项目，加入一些延时处理，以及 User 信息管理等等。



## APPENDIX

---

### Note:

1. This report should be a Microsoft Word or PDF document **no more than 12 pages**.
2. Each student should submit an individual report to **18720429@qq.com** named "**studentID-Name.doc/.pdf**" before **June-15-2019**.
3. If your game has been deployed in a certain app store, it will be appreciated and may get higher final score.

### Fiction Story:

“诸葛亮智算华容，关云长义释曹操”，是世人皆知的一个故事，《三国演义》中的重点篇章。

小说写赤壁战前，诸葛亮算定曹操必败走华容，且夜观天象，曹操不当身亡，考虑到曹操与关羽有恩，于是派关云长把守华容道，留个人情与关羽做。小说又写曹操果然由乌林向华容道败退，并在途中三次大笑诸葛亮、周瑜智谋不足，未在险要处暗设伏兵。

然而，一笑笑出赵子龙，多亏徐晃、张郃二人双敌赵云，才使曹操得以逃脱；

二笑笑出张翼德，又是张辽、徐晃二将抵挡张飞，使曹操再次脱险；

三笑非同小可，笑出了关云长，且又在有一夫当关之险的华容狭路上，加之曹军几经打击，此时已无力再战，无奈，曹操只得亲自哀求关羽放行，关羽念旧日恩情，义释曹操，使曹操得以回到江陵。

后关羽为孙权所杀，人头交由曹操，曹操为其塑木制身体。