本次课题：制作HTML5在线五子棋对弈游戏。

预计开发周期：还没想好看心情，先预计7天完成。

备注：最近做学校课题有关数据挖掘的，有时候搞得没得头绪，做个小游戏和大家分享一下，并将小程序开源到Github，欢迎下载。

分析：

制作在线对弈游戏需要考虑两个部分，前端与后端（废话），动手前先统观全局，逐一布局，免得后期还要大改。先考虑第一部分中的一个点，搭设一个前端游戏界面部分，因为游戏对弈逻辑简单，那么为了方便专注于HTML5游戏开发的小伙伴学习参考，该部分也作为一个独立的小程序（单机五子棋）设计，在实际上五子棋这种小型游戏其实不必提供前端的部分对弈逻辑，但对于更高难度游戏设计而言，提供该部分可以优化用户体验那么我在此也一并设计，由于前端渲染部分并不会与后台有所联系，那么接下来我们就单独好好的研究一下前端的游戏界面设计。因为功能独立，有关其他问题我们来日方长。

本项目单机版五子棋设计仓库地址：（后续可会另创建大课题独立仓库）

<https://github.com/liuxinyumocn/LittleGame>

任务：

设计一个可移植嵌入于网页内部的游戏组件。有朋友会问，为什么要这样设计，好的程序是可以做到能便于移植且功能独立易于扩充，那么就本文设计而言可能多数部分对于新手而言会稍微难于理解，还是希望能够有始有终，仔细理解。

在该课题启动前，我已经做了部分准备可供多数场景复用，在开始前还请各位移驾本人事前所制插件处下载相关控件（原创）。

IAniamtion.js插件下载地址：

<https://github.com/liuxinyumocn/IAnimation>

该仓库中IAniamtion.js与IGraphElement.js即为后文所需文件。

先简单介绍一下IAniamtion.js和IGraphElement.js这两个文件，因为我们需要使用HTML5中Canvas，这两个文件是本人上次参加中国大学生计算机大赛时所自制的小插件，其作品地址<http://basketball.liuxinyumo.cn/>可在线试用其所成效果（进入后无需注册，点击齿轮－常规－新建即可），其功能所在是将Canvas画布中绘制图像进行元素对象化便于管理控制，大致符合Windows窗口控件使用习惯，使用该插件可便于设计且可用于多个基于Canvas应用的场景。具体如何使用我会在用到之处写清，一目了然，尽量满足基本游戏制作需求（其实是我懒的，我有所写该引擎开发者API文档，但实在是乱七八糟不如实战所用，用到即讲）。

IAnimation.js文件是已经被封装好的插件，引用后无需再对该文件进行修改，IGraphElement.js是对场景中所有物景定义，需要编辑。使用时请将IGraphElement.js优先引入。

好了废话也说了不少了，我们动刀子了。先创建一个HTML文件用于提供动画场景，并顺次引入以上下载的两个文件。

<!DOCTYPE html>

<html>

<head>

<title>基于HTML5的在线小游戏制作 Power By Liuxinyumo.cn</title>

</head>

<body>

</body>

<script type="text/javascript" src="IGraphElement.js"></script>

<script type="text/javascript" src="IAnimation.js"></script>

<script type="text/javascript">

//该代码段用于编写游戏部分，但之后我们会放在独立的Game.js文件中以便移植

</script>

</html>

我们先建立一个小的游戏欢迎界面，让我想想……算了，太华丽的界面不是我们本课题重点，就用两个居中的图片信息好了。

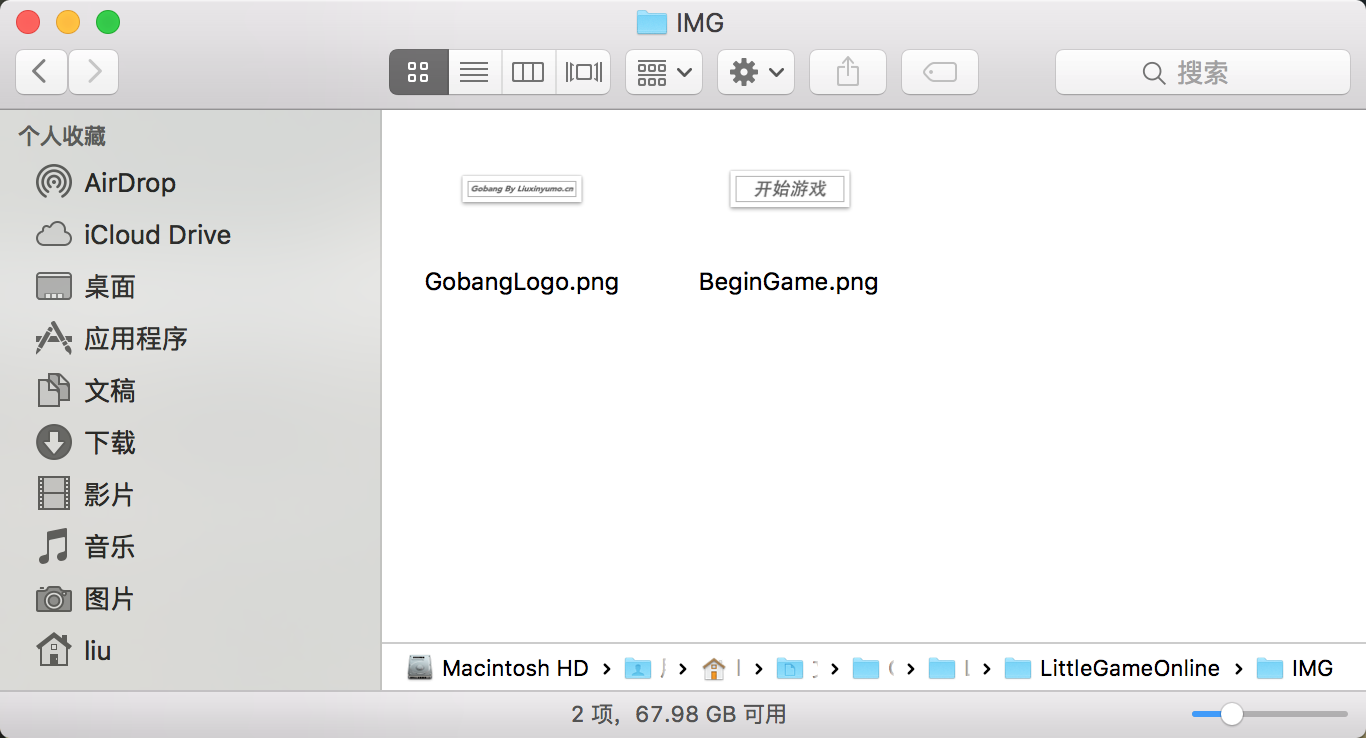
我先设想下大致布局效果……



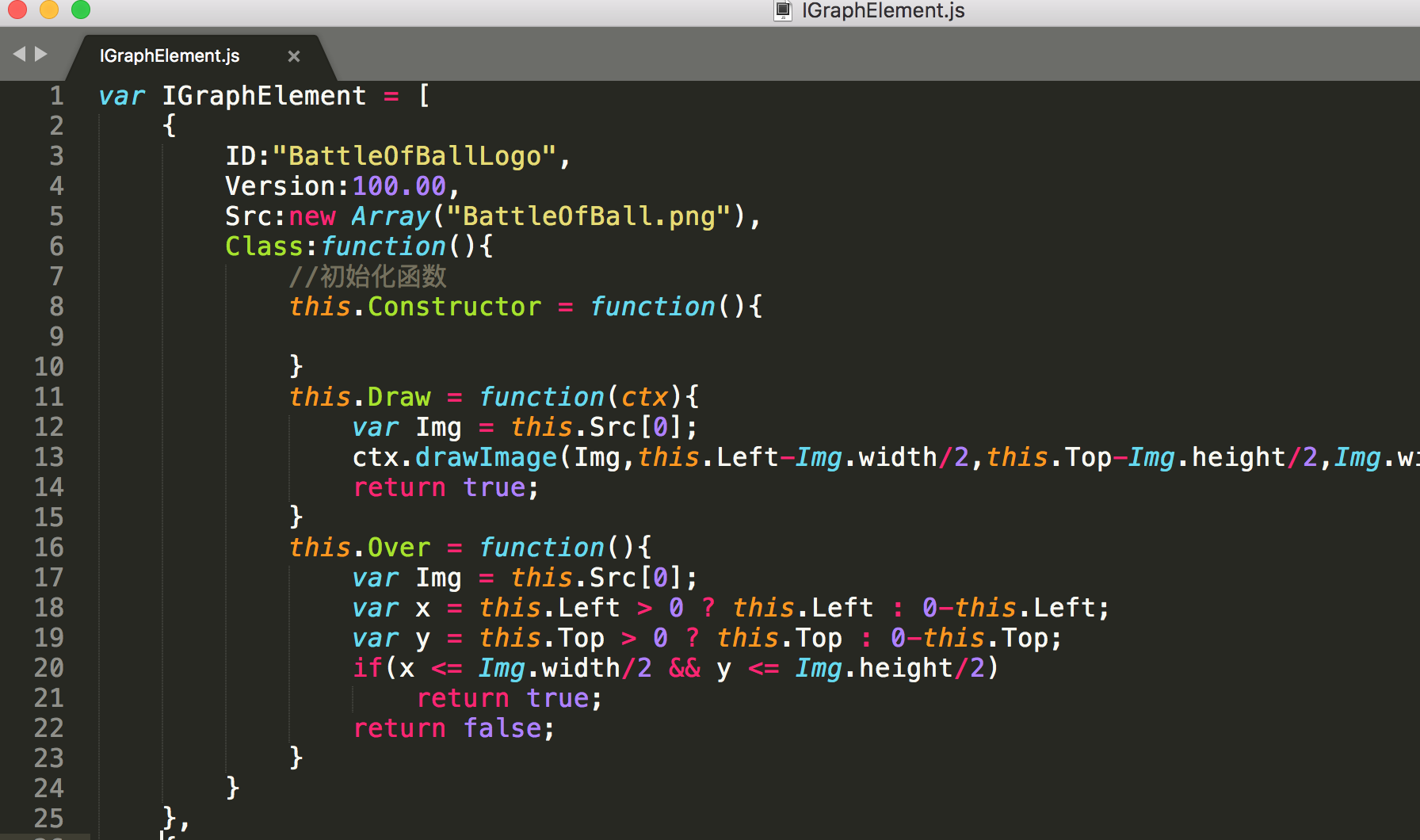
emmmm……真是简单的不能再简单。好了，我们看看这么简单的界面是怎么由Canvas呈现出来的。

第一个场景中我们用到了两个外部图形元素，我们右键使用代码编辑器打开IGraphElement.js文件，来定义一下游戏的欢迎界面中所用到的图像信息，这里我用到了两个图片文件，均放在IMG目录下，可在GitHub仓库中查看。

说明：IMG目录是IGraphElement.js的默认目录，在无冲突的情况下使用默认目录即可，如若冲突使用IAnimation.RootDir("NewDir");函数即可变更。



在IGraphElement.js文件中，我们打开能看见已经写有的部分模板，这是使用模板用于参考，制作新项目时我们需要将其删掉，不过删之前有必要跟大家简单介绍下模板：

可以注意，该文件实际为定义一个数组变量，数组元素为以JSON格式定义的对象信息，我们应以数组追加的格式来扩充此库。先拿第一个元素为例。

我们按代码行编号说话，2-25行为对第一个元素的定义，注意25行因其后还有数组元素，所以用逗号结尾，如果当前元素为数组中最后一个元素，则不应再有逗号。

3行是定义该对象的识别ID，用于创建该对象，类比Windows中诸如Button、Label这样的名词，换句话说，用于实例化该类成为对象个体。

4行用于未来扩展，暂时照写即可。

5行较为关键，这里用于定义该元素所用到的外部资源地址，其默认根目录是IMG，即我们刚刚所提及目录，也可以向根目录下继续追加（项目足够大的话是可以划分资源文件目录）例如“abc/Battleofball.png”则表示：IMG/abc目录下文件。

6-24行是定义元素动作及属性，换言之，这里就是在定义一个类。具体的如下描述：

8行所定义是构造函数（必须），因当前例子中没有成员变量，因此构造器为空的，构造器用于初始化变量，变量定义中不可直接赋值应在构造器中初始化，确保引擎调用时正常工作。

11行所定义是绘制函数（必须），这里需携带一个ctx参数，ctx参数是Canvas的绘图句柄，换言之，此处即为对该对象所呈现图形的具体描述定义，需使用底层CanvasAPI绘制，这里注意一下12行，12行也是本引擎的一个内置用法，this.Src[0]实际上就是外部资源，即5行代码中第一个元素，利用该引擎可以在应用启动前统一进行资源加载，也就是你们玩flash游戏也会看见downloading这样的百分比字样，因为要统一异步加载资源保持程序流畅，并且提供更友好的界面体验，这就是为什么我们要有第5行的原因。到这里，this.Src[x]（x是对应数组索引）就直接拿来当图片资源使用即可，程序如若运行到此，资源一定加载完毕。

16行所定义的是拾取算法，这里会有两个坐标值，获取方法使用形如12行，x坐标为this.Left，y坐标为this.Top。注意，无论是x还是y，使用this.Left/Top时，只能用于获取该值，不能修改该值（这是因为js没有变量指针），如若修改则应该使用Self方法，使用方法为：this.Self.Left(NewValue);该方法会修改该元素的绝对坐标。有关绝对、相对坐标我们先不谈，因为这样势必有要一大篇幅，因为我的引擎不仅仅适用于小型项目，在跨尺寸移植时会涉及到屏幕适应，因此我的引擎内部会有两套坐标在工作，有兴趣的朋友我再另述。在拾取块中你需要利用已知的xy坐标，和标准坐标系下（假设你的影像没有被缩放或旋转且你所定义的左上角在坐标圆点）给出自定义图像是否被选中。

16行的拾取算法是否太过绕人？不妨我们实战使用，其实我写的引擎会为各位提高足够高的工作效率，当你真正理解之后才会明白，因为你真的无须考虑该元素此时处于何地，缩放了没有，旋转了没有，一切按照最理想的状态判断即可。

还拿该模板为例：11-13行实际上做了一件非常简单的事情，将一张图片绘制出来，绘制的坐标为 (this.Left-Img.width/2, this.Top-Img.height/2)，也就是说，我所希望的元素的Top、Left描述的是图片的中心，因为绘制坐标一定是从左上角开始，而我希望管理该图片的坐标是图片的中心，你们仔细理解一下。那么拾取部分我并没有在意这个元素实际上在哪，是否旋转，而是尽管假设把它放在了原点，没有被旋转缩放，拿this.Left/Top是否超越图片矩形区域，来判断能否拾取到，如若拾取到则return true; 反之return false;各位伙伴这里我就不再冗余介绍，还需自行理解。