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。

接上期。传送门：

<http://liuxinyumo.cn/index.php?s=/Home/Index/content/id/3.html>

上期我们有介绍IGraphElement.js到底该怎么定义元素，接下来继续设计那个草率的界面。

继续使用代码编辑器打开IGraphElement.js，删除掉模板元素，或者留有几个元素，在该模板上进行修改。

如下代码：

var IGraphElement = [

{

ID:"GobangLogo",

Version:100.00,

Src:new Array("GobangLogo.png"),

Class:function(){

//初始化函数

this.Constructor = function(){

}

this.Draw = function(ctx){

var Img = this.Src[0];

ctx.drawImage(Img,this.Left-Img.width/2,this.Top-Img.height/2,Img.width,Img.height);

return true;

}

this.Over = function(){

var Img = this.Src[0];

var x = this.Left > 0 ? this.Left : 0-this.Left;

var y = this.Top > 0 ? this.Top : 0-this.Top;

if(x <= Img.width/2 && y <= Img.height/2)

return true;

return false;

}

}

},

{

ID:"BeginGameButton",

Version:100.00,

Src:new Array("BeginGame.png"),

Class:function(){

//初始化函数

this.Constructor = function(){

}

this.Draw = function(ctx){

var Img = this.Src[0];

ctx.drawImage(Img,this.Left-Img.width/2,this.Top-Img.height/2,Img.width,Img.height);

return true;

}

this.Over = function(){

var Img = this.Src[0];

var x = this.Left > 0 ? this.Left : 0-this.Left;

var y = this.Top > 0 ? this.Top : 0-this.Top;

if(x <= Img.width/2 && y <= Img.height/2)

return true;

return false;

}

}

}

];

按照我之前的讲解，这些代码并不是很难理解，大致都是在做一件事，分别讲我们之前的两个图片，以中心为控制坐标点定义出来。

定义之后，我们打开我们之前所做的HTML文件，我在之前的HTML文件引用的两个script的下面有注释，另创建Game.js文件单独存放游戏部分代码，也就现在创建Game.js，并在该处引入到HTML文件，我们这里将HTML文件保存为Gobang.html。

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

<script type="text/javascript">

//用于移植的控制代码

</script>

使用代码编辑器打开Game.js，为了开发标准化防止变量泄露，我们创建一个js的自执行函数，并对外暴露一个Gobang的游戏类名。在Game.js文件中写入：

var Gobang;

(function(){

Gobang = function(Container){

//Gobang游戏类的构造函数

}

Gobang.fn = Gobang.prototype = {

}

})();

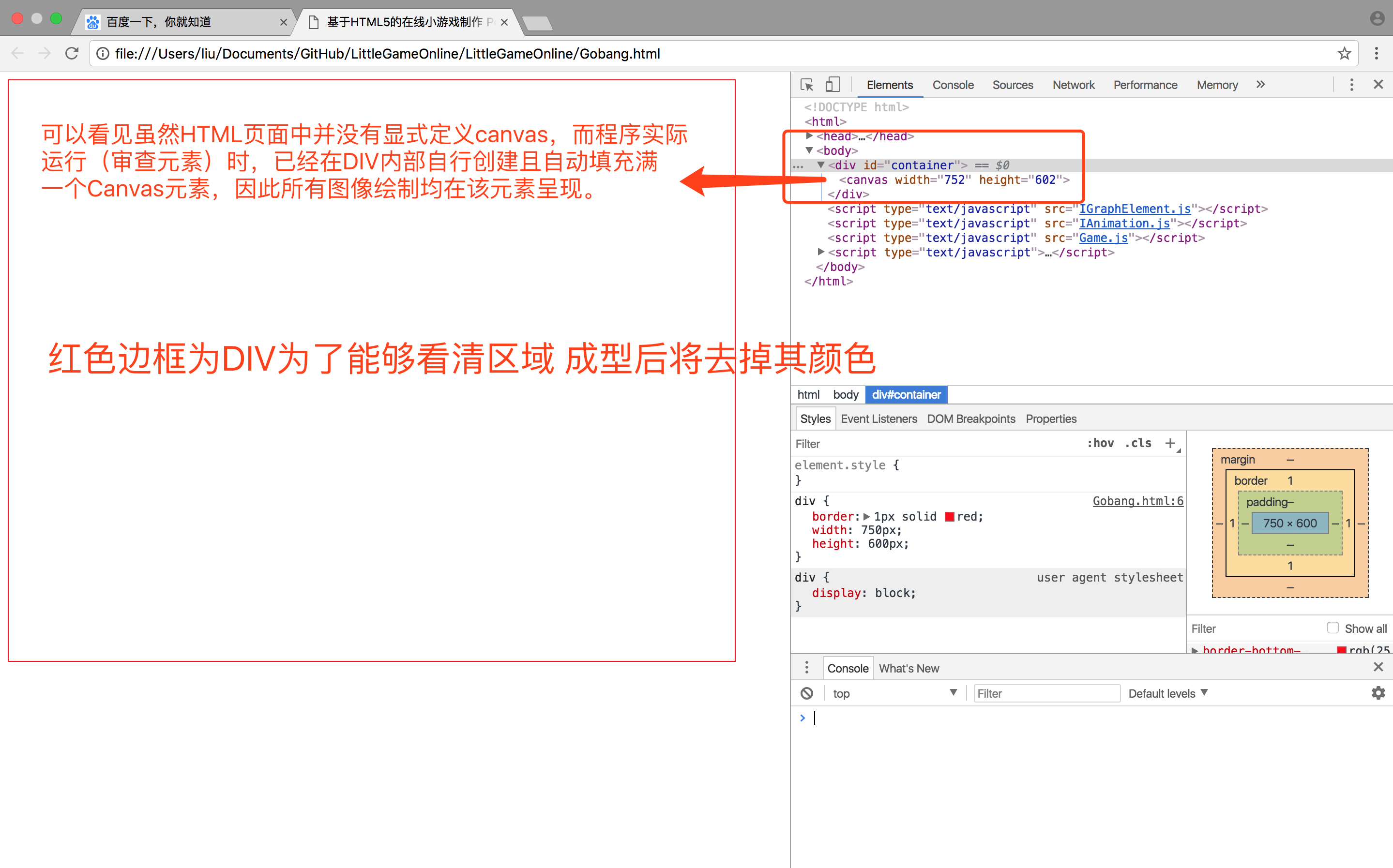
在构造函数中提供1个参数，用于传递存放Canvas画布的容器句柄，IAnimation.js插件并不是需要我们显式提供<canvas></canvas>元素，而是提供容器，自行填充Canvas，填充的大小会自动充满整个容器内部，这样做的一大好处是大大方便了项目移植。

这里注意我们在编辑Gobang的类时，所有的成员变量要定义成 this.xxx，所有的成员函数应定义成 this.xxx= function(){}，成员函数的定义也可以在prototype中使用形如JSON的定义方式，但变量数据不能在该处定义。这里我们约定所有函数使用首字母大写命名，对私有成员的标注我们在命名前增加一个下划线区分。

接下来我们在Gobang构造器中初始化IAnimation，写入如下代码：

this.\_Animation = new IAnimation(Container); //Container是参数

我们回到Game.html页面中提供一个Div容器，并使用CSS控制其大小，然后在用于移植的控制代码段中实例化Gobang类，随后我们打开浏览器查阅下执行效果。



给一下完整的Gobang.html代码内容：

<!DOCTYPE html>

<html>

<head>

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

<style type="text/css">

div{

border: 1px solid red;

width: 750px;

height: 600px;

}

</style>

</head>

<body>

<div id="container"></div>

</body>

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

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

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

<script type="text/javascript">

//用于移植的控制代码

var Container = document.getElementById("container");

var game = new Gobang(Container);

</script>

</html>

那么现在有了游戏视窗，我们将我们第一个场景，欢迎页面布置上。由于该景布置归属于游戏自身，因此我们打开Game.js在该文件中进行定义。

在使用自定义元素前我们需要先对外部资源进行选择性的异步加载，这里是一个扩充数组，将我们所有用到的自定义元素ID进行列举，告诉引擎哪些是本次作业需要加载的资源项。通用的写法为：

var t = this;

function CallBack(a,b){

if(a == b){

t.\_DownloadFinished();

}

}

this.\_Animation.DownloadIMGing = CallBack; this.\_Animation.DownloadIMG(["GobangLogo","BeginGameButton"]);

我们需要将该代码段单独放到一个 *Init()的函数里，同时对Gobang创建一个*\_DownloadFinished()函数。

\_DownloadFinished()函数一旦之行到意味着我们想要加载的外部资源已经准备就绪，因此我们就可以调用自定义资源库中的元素了，我们在初始化阶段实际上是可以将游戏界面、欢迎界面一并装在，游戏界面我们随后设计，因此在代码结构上留好位置。

接下来开始布局欢迎界面，单独定义创建函数\_CreateWelcomeScene()，引擎中我们使用Page来区分场景，因此首先利用引擎创建一个WelcomePage，并保存到成员函数中，将该名称在构造器中声明，以便日后查找。使用this.\_Animation.CreatePage() 即可创建一个场景，返回值赋值给我们所定义的this.\_WelcomePage变量。接下来我们就可以利用该变量向场景中添加自定义的元素了。

利用this.\_WelcomePage.AddElement("GobangLogo");即可创建Logo元素，但元素默认Visible属性均是false，所以先对其进行位置布局，再将其设置为可视。同理，我们对BeginGame也做同样的处理。

保存后我们运行下效果：



还算满意，给下Game.js到目前为止的完整代码。

var Gobang;

(function(){

Gobang = function(Container){

//Gobang游戏类的构造函数

this.\_Animation = new IAnimation(Container);

//成员变量声明

this.\_WelcomePage = null;

this.\_Init();

}

Gobang.fn = Gobang.prototype = {

\_Init:function(){

var t = this;

function CallBack(a,b){

if(a == b){

t.\_DownloadFinished();

}

}

this.\_Animation.DownloadIMGing = CallBack;

this.\_Animation.DownloadIMG(["GobangLogo","BeginGameButton"]);

},

\_DownloadFinished:function(){

//此处外部资源已经全部加载完毕可以创建游戏场景。

this.\_CreateWelcomeScene();

//这里留有部分用于创建游戏界面

},

\_CreateWelcomeScene:function(){

//创建游戏欢迎页面

this.\_WelcomePage = this.\_Animation.CreatePage();

//在场景中布局

var Logo = this.\_WelcomePage.AddElement("GobangLogo");

Logo.Left(this.\_Animation.Width/2);

Logo.Top(this.\_Animation.Height/2-100);

Logo.Visible(true);

var BeginGameButton = this.\_WelcomePage.AddElement("BeginGameButton");

BeginGameButton.Left(this.\_Animation.Width/2);

BeginGameButton.Top(this.\_Animation.Height/2+20);

BeginGameButton.Visible(true);

}

}

})();

篇幅也已经很长了，本篇先结束。