



SuperSuRaccoon

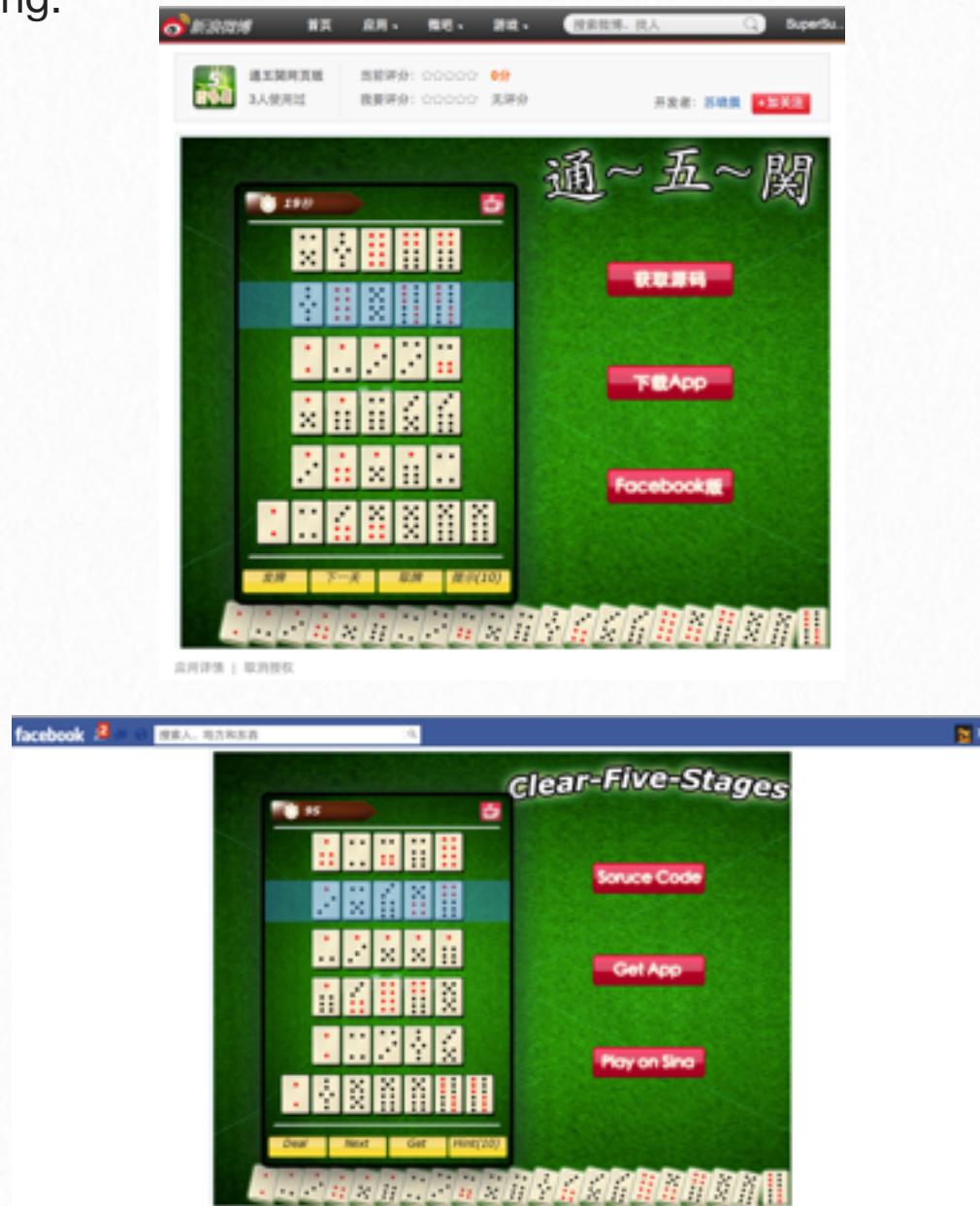
A trip from *cocos2d-iphone* to *cocos2d-html5*



1

The start - Porting an ios game to html5

The reason why I write this article is pretty simple - porting one of my former game on ios using cocos2d-iphone to web using cocos2d-html5 and share what I've learnt from the whole process to the ones who might be interested in or are planning to the very same thing.



Introduction to cocos2d-html5

Being the most famous mobile 2D game engine, there are also lots of branch versions of cocos2d, and one of the hottest version right now will be -- the cocos2d-html5.

And here is a brief introduction from cocos2d-x home page:

"Cocos2d-html5 is an open-source web 2D game framework, released under MIT License. It is a HTML5 version of Cocos2d-x project. Our focus for Cocos2d-html5 development is around making Cocos2d cross platforms between browsers and native application.

On top of the framework provided by Cocos2d-html5, games can be written in Javascript, using API that is COMPLETELY COMPATIBLE with that of Cocos2d-iPhone, Cocos2d-x javascript binding.

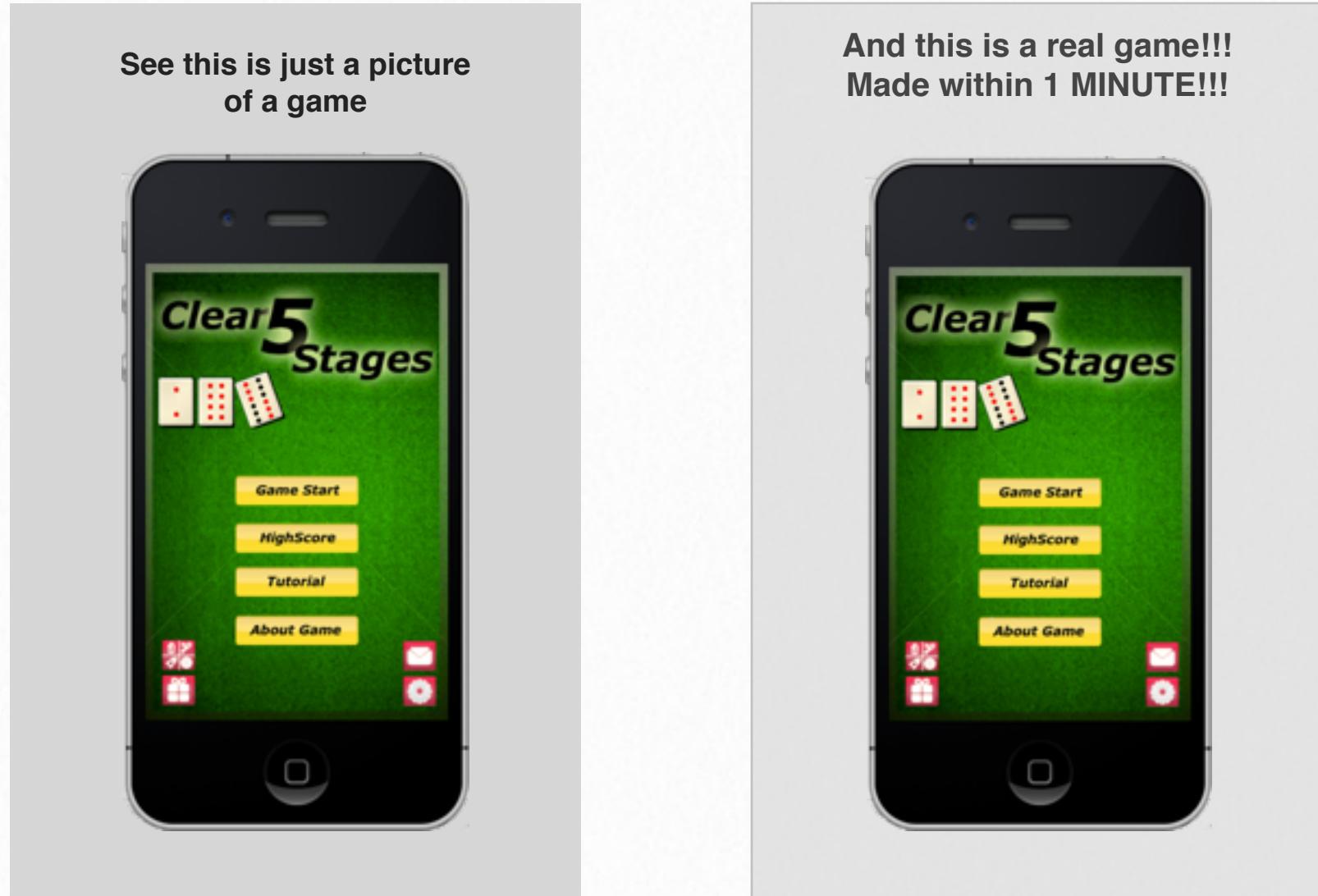
Cocos2d-html5 project can be easily run on browsers which supports HTML5. Cocos2d-html5 also supports running game code in "Cocos2d Javascript binding" without or with little modification."



Why html5 ?

I believe that every game developer has his own opinion on the perspective of HTML5 game but regardless of all that theory there is a simple but realistic advantage of HTML5 game which is that there is a lot of platforms you can deploy your game on and that means more users, more exposure and more profit.

And HTML5 is full of magic -- check out the two “pictures” below:



Why cocos2d-html5 ?

Actually this might be the easiest decision to make since it is doom to be the right choice with the great great cocos2d-x community as our powerful backup.

And for those developers who have started from the cocos2d-iphone period -- I think me myself is a quite good example out of this -- it is even easier to arm your self with a new weapon that is called -- cocos2d-html5.



Something about this book

I'm somewhat familiar with cocos2d-iphone but when it comes to the cocos2d-html5 --

I must confess -- I've never had any experience with it and javascript, html, css and php before writing this article as well.

But I think there is both pros and cons of this fact:

The pro's:

- Only a beginner who have just learnt his way through a lot of difficulties will have the ability to write something that is perfect for the beginners 😊
- I think that's what you'll find in this book -- simple as ABC but practical -- all the stuffs come from the real experience.
- A good evidence of learning cocos2d-html5 is easy especially for those have cocos2d-iphone experience.

The con's:

- There gonna be some flaws and even errors in this book since just like I mentioned I'm a complete rookie on javascript, html, css and php that we are gonna use through out the whole book 😢

OK, time for the trip ~ grab yourself a cup of coffee and fasten your seat bell - it's

gonna be a bumpy ride 😎

2

Tools and knowledge - A summary on what we need for each step

Great tools are always helpful when developing games even for veterans.

Since this book also talks about stuffs on debug and deploy there are quite a lot of tools will be used and knowledge of course through out the whole trip.

And here is a summary of what tools we are gonna use and what specific knowledge we'd need to make our way to the goal.

A checklist of the what we need

Tool list:

1. xcode 4.6 -- no comment 😊
2. cocos2d-iphone v2.1-beta4 -- cocos2d iphone version, watch out for the version here
(2nd coordinated release)
3. cocos2d-html5 v2.1 - cocos2d html5 version **(2nd coordinated release)**
4. cocosBuilder v3.0-alpha0 -- visual editor for cocos2d **(2nd coordinated release)**
5. TexturePacker -- create sprite sheets for our game
6. Jetbrains WebStrom -- the most intelligent JavaScript IDE
7. WAMPServer -- a web development platform on Windows allowing you to create dynamic web applications with Apache2, PHP and MySQL under windows
8. MAMP -- WAMPServer mac version
9. Google Chrome -- browser for debug and testing
10. FireFox -- browser for debug and testing
11. FireBug -- tools for web development. Allows inspect, edit and monitor CSS, HTML, JavaScript
12. textastic -- a text editor for iPad and iPhone with rich support for syntax highlighting, FTP, SFTP and Dropbox
13. heroku toolbelt -- heroku command line tool

14. git -- git command line tool

15. CopSSH -- an OpenSSH server and client implementation for Windows systems

16. TortoiseSVN -- SVN tool for Windows OS

17. SmartSVN -- SVN tool for Mac OS

Knowledge list:

1. Objective-c -- needed for ios develop

2. javascript -- needed for html5 game develop

3. html -- needed for understanding our index.html

4. php -- needed for game deployment and game oath

5. git command -- needed for game deployment

6. css -- needed for creating a good looking home page for our game

If you feel a little bit frustrated by the long list then I can tell you it's nothing to worry about since we do not need to be a expert on each tool and knowledge, we only need to learn some basic but enough for porting our game and distribute it to the world and that's really not a hard task.

Well, sip of coffee and time to port out game to HTML5 

3

Basic stuffs - Porting to cocos2d-html5

For those who are familiar with cocos2d-iphone version but seldom use the cocos2d-x like myself will probably get confused at the beginning.

We use designed initializer a lot in cocos2d-iphone say "spriteWithFile" when creating a CCSprite but in cocos2d-x and html5 version we use the "create" function a lot.

Anyway, it's just a matter of function name, the meaning and sequence of parameters are almost the same so it won't be hard to get familiar with.

Comparison between cocos2d-iphone against

```
var aSprite = cc.Sprite.create("xxx.png");
this.addChild(aSprite, 1, 1);
```

Let's begin with some very familiar stuffs:

- The class name:

Basically changing all the CCXXXX in iphone to cc.XXXX will just do the trick:

```
CCSprite -- cc.Sprite
CCFadeOut -- cc.FadeOut
CCLabelTTF -- cc.LabelTTF
```

but you should also watch out for some special case like something in ccType.h(js) and CCGeometry.js:

```
ccWHTIE -- cc.white (watch out for the Ucase/Lcase letters)
ccp -- cc.p
CGRectContainsPoint -- cc.Rect.CCRectContainsPoint
```

And then let's check out some basic code snippets:

- Creating CCSprite:

cocos2d-iphone:

```
CCSprite *aSprite = [CCSprite spriteWithFile:@"xxx.png"];
[self addChild:aSprite z:1 tag:1];
```

cocos2d-html5:

```
var aSprite = cc.Sprite.create("xxx.png");
this.addChild(aSprite, 1, 1);
```

- Creating CCSprite using spritesheet:

cocos2d-iphone:

```
[[CCSpriteFrameCache sharedSpriteFrameCache]
addSpriteFramesWithFile:@"xxx.plist"];
CCSprite *aSprite = [CCSprite spriteWithSpriteFrameName:@"xxx.png"];
[self addChild:aSprite z:1 tag:1];
```

cocos2d-html5:

```
cc.SpriteFrameCache.getInstance().addSpriteFrames("xxx.plist", "xxx.png");
var aSprite = new cc.Sprite();
aSpriteinitWithSpriteFrameName("xxx.png");
this.addChild(aSprite, 1, 1);
```

- Simple actions usage (watch out for the "create" function in html5 version):

cocos2d-iphone:

```
[self runAction:[CCRepeatForever actionWithAction:[CCJumpBy
actionWithDuration:1.0f position:ccp(0, 0) height:10 jumps:1]]];
```

cocos2d-html5:

```
this.runAction(cc.RepeatForever.create(cc.JumpBy.create(1.0, cc.p(0, 0), 10, 1)));
```

As you can see it's quite easy to port those basic stuffs to cocos2d-html5.

I think it can be boil down to:

- Replace the "self" keyword with "this"
- Replace the "[]" function calling style with "()"
- Replace the "CCXXXX" style class Designed initializer "cc.XXXX" (watch out for special stuffs)
- Replace the "Designed initializer" with the "create" function 

On porting CallFunc to cocos2d-html5

The CallFunc serial functions are the most frequently used ones in cocos2d-iphone.

It's a little bit different when it comes to cocos2d-html5 since there is only one function named "cc.CallFunc" in comparison with 3 the functions "CallFunc, CCCallFuncN, CCCallFuncND" in cocos2d-iphone and what's more the meaning of the 1st and 2nd parameters are different -- it's [target, selector] in cocos2d-iphone while [callback, target] in cocos2d-html5.

cocos2d-iphone:

```
[self runAction:[CCCallFuncND actionWithTarget:self  
selector:@selector(someFunc:data:) data:(void *)someParam]];
```

cocos2d-html5:

```
this.runAction(cc.CallFunc.create(this.someFunc, this, someParam));
```

And then the selector definition:

cocos2d-iphone:

```
-(void) someFunc:(id)sender data:(void*)data {}
```

cocos2d-html5:

```
someFunc:function (sender, data) {}
```

On porting Array and Dictionary to cocos2d-html5

The last thing we're gonna talk about is the array and dictionary in the cocos2d-iphone
-- CCArray , NSMutableArray and NSMutableDictionary.

At first I thought there is something similar to CCArray , NSMutableArray or NSMutableDictionary in cococs2d-html5 but after looking into the code I know there isn't.

Then I did little bit search on the array and dictionary in javascript and found there is an "Array" object in javascript but no Dictionary which BTW we can use the "Array" to simulate.

The usage of array in javascript is quite straightforward:

cocos2d-html5:

```
var aArray = new Array();
aArray.push(1);
aArray.push(2);
aArray.push(3);
for (var i = 0; i < aArray.length; i++) cc.log(aArray[i]);
```

but it seems to be lack of functions such as "insertAt", "removeAt" etc.

Anywhere, we can implement there features easily with the flexibility of javascript and here are some solutions I found from the googling:

- Clear Array:

```
Array.prototype.clear=function(){
    this.length=0;
}
```

- Insert object at a give position in array:

```
Array.prototype.insertAt=function(index,obj){
    this.splice(index,0,obj);
}
```

- Remove the object at given position in array:

```
Array.prototype.removeAt=function(index){
    this.splice(index,1);
}
```

- Remove give object in array:

```
Array.prototype.remove=function(obj){
    var index=this.indexOf(obj);
    if (index>=0) this.removeAt(index);
}
```

- Check if the array contains the given object:

```
Array.prototype.contains = function(obj) {  
    var i = this.length;  
    while (i--) if (this[i] === obj) return true;  
    return false;  
}
```

As for the dictionary in javascript we can simulate with the array object as mentioned above:

```
aArray['name'] = aName;  
aArray['age'] = aAge;  
aArray['fax'] = aFax;
```

But there's one thing you should notice that the such array is called the "associative array" and if you try to use the ".length" function to calculate the length of the dict in a for loop you will get the wrong result and there is a solution for this:

```
Object.size = function(obj) {  
    var size = 0, key;  
    for (key in obj) if (obj.hasOwnProperty(key)) size++;  
    return size;  
};
```

How to use it:

```
for (var i = 0; i < Object.size(aArray); i++) {...}
```

and here is a post on stackoverflow you can checkout:

<http://stackoverflow.com/questions/5223/length-of-javascript-object-ie-associative-array>

As you can see from above it is really easy to port the basic stuffs from cocos2d-iphone to cocos2d-html5 but when it comes to the special stuffs it is not as simple as replacing some function names.

But don't worry since we are gonna cover those special stuffs in the next chapter.

One more sip of coffee and take a breath and let's get on the road again



4

Special stuffs - Porting to cocos2d-html5

But what on earth are the special stuffs here ?

Well, it's something that not in "code snippet" level but "code structure" or "code characteristics" level. So what we are gonna talk about is how to implement the "singleton pattern", "delegate pattern" and "the use of block" in cocos2d html5.

Before we get started the first thing we have to learn is a basic but important concept in javascript (for beginners).

On “function” in cocos2d-html5

The important concept we are gonna talk about is "function".

Since there is no such keyword as "class" in javascript -- which you might already know

-- we need to use the "function" to implement all of our "class".

- Using function to simulate class:

```
var SpriteClass = function () {};
```

- Adding member variable to class:

```
var SpriteClass = function () {
    this.memberVariable = 1;
};
```

- Adding member function to class:

```
var SpriteClass = function () {
    this.memberVariable = 1;
    this.memberFunction = function() {};
};
```

- Adding member function to class: (using property to avoid function copy):

```
var SpriteClass = function () {  
    this.memberVariable = 1;  
};  
  
SpriteClass.property.memberFunction = function () {};
```

- Adding class variable to class:

```
var SpriteClass = function () {  
    this.memberVariable = 1;  
};  
var SpriteClass.classVariable = 2;
```

- Adding class function to class:

```
var SpriteClass = function () {  
    this.memberVariable = 1;  
};  
SpriteClass.classFunction = function () {};
```

Based on the knowledge above here is a simple example showing how to subclass a CCSprite and then add a "create" function to it in cocos2d-html5:

```
var ChildSprite = cc.Sprite.extend({
    aProperty:null,
    bProperty:null,
    init:function (a, b) {
        var bRet = false;
        if (this._super()) {
            this.aProperty = a;
            this.bProperty = b;
            bRet = true;
        }
        return bRet;
    }
});

ChildSprite.create = function (a, b) {
    var childSprite = new ChildSprite();
    if (childSprite && childSprite.init(a, b)) return childSprite;
    return null;
};
```

If you're familiar with cocos2d-iphone you might notice that the code flow is almost the same in cocos2d-iphone



On Singleton in cocos2d-html5

We are using the Singleton a lot in cocos2d-iphone -- "[XXXX sharedXXXX]" and in the cocos2d-html5 it's all replaced with the "xxxx.getInstance()". And the implementation is quite easy since we already learnt the knowledge we need in the former section.

How to use:

cocos2d-iphone:

```
CGSize winSize = [CCDirector sharedDirector].winSize;
```

cocos2d-html5:

```
var winSize = cc.Director.getInstance().getWinSize();
```

A simple implementation in cocos2d-html5:

```
GameManager.getInstance = function () {
    if (!this._sharedGame) {
        this._sharedGame = new GameManager();
        if (this._sharedGame.init()) return this._sharedGame;
    } else {
        return this._sharedGame;
    }
    return null;
};
GameManager._sharedGame = null;
```

And you can also look into the cocos2d-html5 source since there are lots of Singleton there 😊

On Delegate in cocos2d-html5

We use a lot of "delegate" in cocos2d-iphone and actually it's pretty simple to implement in cocos2d-html5.

Defining and calling delegate in class:

```
var GameManager = function () {
    this.delegate = null;
};

GameManager.prototype.doSomething = function () {
    if(this.delegate) this.doSomethingDelegate();
};
```

Initializing delegate:

```
var g = new GameManager();
g.delegate = this;
XXXLayer.prototype.doSomethingDelegate = function () {}
```

Piece of cake, right ? 

On Timer in cocos2d-html5

Another useful concept in cocos2d-iphone must be the timer which we can use to do lots of things.

- Defining and calling simple timer:

cocos2d-iphone:

```
-(void) timer:(ccTime)dt {}  
[self schedule:@selector(timer:) interval:1.0f];
```

cocos2d-html5:

```
XXXScene.prototype.timer = function (dt) {};  
this.schedule(this.timer, 1.0);
```

- Timer delay in cocos2d-html5:

In cocos2d-iphone we sometimes use the “perform selector” to achieve some effect we could not done with CallFunc and we can also do the same thing in cocos2d-html5 using setTimeout.

- Simple Delay timer:

cocos2d-iphone:

```
[self performSelector:@selector(someSelector) withObject:self afterDelay:1.0f];
```

cocos2d-html5:

Watch out for the difference in time unit and we'll talk about the “**that**” in the later chapter you can consider it as same as the “this” keyword at this time.

```
var that = this;
setTimeout(function(){that.someSelector();}, 1.0 * 1000);
```

- Delay timer with parameter:

cocos2d-iphone:

```
[self performSelector:@selector(someSelector:) withObject:param
afterDelay:1.0f];
```

cocos2d-html5:

```
var that = this;
(function(param){setTimeout(function(){that.someSelector(param);}, 1.0 *
1000);})(param);
```

On NSSelectorFromString in cocos2d-html5

There is a magic function in objective-c which can turn a string to a selector which we can use to add great dynamic features to are game and in javascript we can also get that done.

cocos2d-iphone:

```
[CCCallFuncND actionWithTarget:self  
selector:NSSelectorFromString(@"selectorName") data:(void *)params];
```

cocos2d-html5:

There is a function named “eval” in javascript which can turn a string to actual code:

http://www.w3schools.com/jsref/jsref_eval.asp

And we can use it to make something similar to “NSSelectorFromString” in javascript.

```
cc.CallFunc.create(eval("this." + "selectorName"), this, params);
```

On block in cocos2d-html5

“Block” is yet another concept we often see and use in cocos2d-iphone and we can use the “Anonymous function” in javascript as a substitution.

Say we got a function to save our game score and we need output the result after done saving:

```
ScoreManager.saveScoreData = function (scoreData, callback) {  
    // do stuffs  
    callback(result);  
};
```

and we can pass a Anonymous function to “saveScoreData” while calling:

```
ScoreManager.saveScoreData(aScoreData, function (result) {cc.log(result);});
```

Now what we have learnt up till now could help us to port 80% of the codes in cocos2d-iphone to cocos2d-html5 as for the left 20% you have to figure them out yourself 😞

Now bottom up and take away your coffee since we are gonna talk about how to use cocosbuilder with cocos2d-html5 and believe me there are lots of traps so be careful 😎

5

Time for cocosbuilder - One tool two versions

Using cocosbuilder when developing with cocos2d game engine has become common knowledge nowadays since it not only can help us simplify the UI develop but also can be used as a great animation tool not mention that the latest version of cocosbuilder already supports cocos2d-html5.

This means that you can design almost everything related to UI and animation and then publish the result to use in different branch of cocos2d family, that's want we call “Write once, run anywhere”.



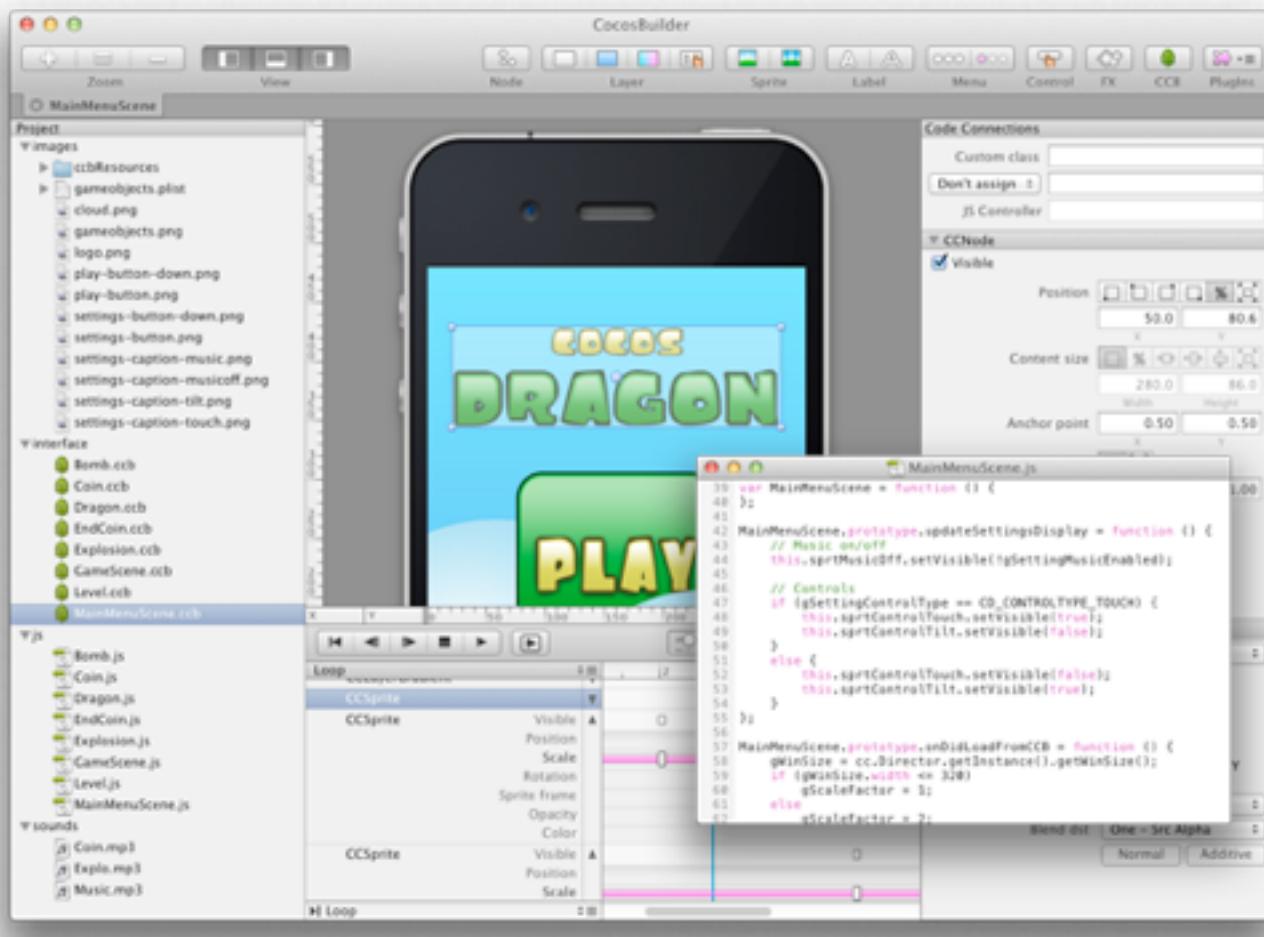
Introduction to cocosbuilder

For those who have started cocos2d develop in the early stage might have experienced the hell of developing with UI stuffs -- layout, positioning, moving pixel by pixel, compile, run and start all over again till satisfied.

But the cocosbuilder has brought heaven to UI design in cocos2d 

Here is the introduction on cocosbuilder homepage:

“CocosBuilder is a free tool (released under MIT-licence) for rapidly developing games and apps. CocosBuilder is built for Cocos2d’s Javascript bindings, which means that your code, animations, and interfaces will run unmodified on iPhone, Android and HTML 5. If you prefer to go native all the way, there are readers available for cocos2d-iphone and cocos2d-x.”



Comparison between cocos2d-iphone against cocos2d-html5 using cocosbuilder

It is quite different using ccbi in cocos2d-html5 compared with in cocos2d-iphone and here I'll show you some basic usage in both versions so that you can see the difference easily.

- Loading scene from ccbi file:

Now watch out for the function name since they're not the same between two versions and if overlook this in the javascript you're gonna cry for this (instead of giving you error for mistaking the function name javascript will create it on the fly thus your scene will never be initialized)

cocos2d-iphone:

```
-(void) didLoadFromCCB {}
```

cocos2d-html5:

```
XXXScene.prototype.onDidLoadFromCCB = function () {}
```

- Creating scene from ccbi:

cocos2d-iphone:

```
CCScene* scene = [CCBReader  
sceneWithNodeGraphFromFile:@"LevelSelectScene.ccbi"];  
[[CCDirector sharedDirector] replaceScene:scene];
```

cocos2d-html5:

```
var scene = cc.BuilderReader.loadAsScene("LevelSelectScene.ccbi");  
cc.Director.getInstance().replaceScene(scene);
```

- Using animation from ccbi:

cocos2d-iphone:

```
CCNode *readySetGoNode = [CCBReader  
nodeGraphFromFile:@"ReadySetGo.ccbi"];  
[self addChild:readySetGoNode];  
CCAnimationManager *animationManager = readySetGoNode.userObject;  
[animationManager setCompletedAnimationCallbackBlock:^(id sender){[self re-  
moveSpriteAndBegin];}];  
[animationManager runAnimationsForSequenceNamed:@"ReadySetGo"];
```

cocos2d-html5:

```
var readySetGoNode = cc.BuilderReader.load("ReadySetGo.ccbi");  
this.rootNode.addChild(readySetGoNode);  
readySetGoNode.animationManager.runAnimationsForSequenceNamed("ReadySet  
Go");  
readySetGoNode.animationManager.setCompletedAnimationCallback(this,  
this.removeSpriteAndBegin);
```

- Binding ccbi to a scene:

cocos2d-iphone:

We need to define a class the same name as we specified in cocosbuilder (custom class).

```
@interface MainMenuScene : CCLayer {}  
@end  
  
@implementation MainMenuScene  
-(void) didLoadFromCCB {}  
@end
```

cocos2d-html5:

Since there's no "class" in javascript we need to use "function" to define a "scene".

```
var MainMenuScene = function () {};  
MainMenuScene.prototype.onDidLoadFromCCB = function () {};
```

- Binding ccbi UI element to variable:

cocos2d-iphone:

All we need is add the variable definition to our interface section with the name we specified in cocosbuilder and then we can use it.

```
CCLabelTTF *helpButtonLabel;  
[helpButtonLabel setString:STRING_GAME_TUTORIAL];
```

cocos2d-html5:

We can use the variable directly without defining it.

```
this.helpButtonLabel.setString(STRING_GAME_TUTORIAL);
```

- Binding ccbi selector to function:

cocos2d-iphone:

We only need to make sure the name is same as specified in cocosbuilder.

```
-(void)onOptionPressed {}
```

cocos2d-html5:

Same here and we can use the prototype mode.

```
XXXScene.prototype.onOptionPressed = function () {}
```

Now we have learnt how to use the ccbi generated by cocosbuilder in both cocos2d-iphone and cocos2d-html5 but I need tell you that things are not that simple.

There are quite a lot of traps and special points when using cocosbuilder along with cocos2d-html5 and we are gonna cover those in the next two chapter (make sure you have put away your coffee cup since I don't want you to smash your computer with it)



6

Watch out for traps - what we should not overlook when using cocosbuilder

The cocosbuilder which supports cocos2d-html5 is still in alpha version while writing this book so it is understandable that there might some bugs or issues and I am gonna share some problems that had while using cocosbuilder with cocos2d-html5 and hope that can help the ones who are having the same trouble

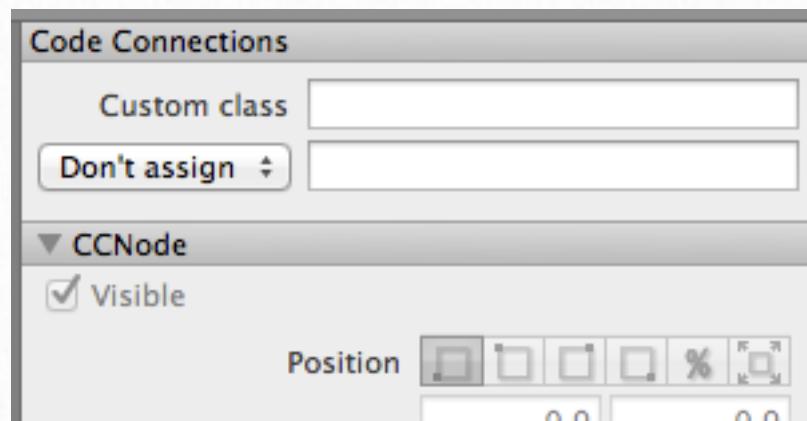


All kinds of traps

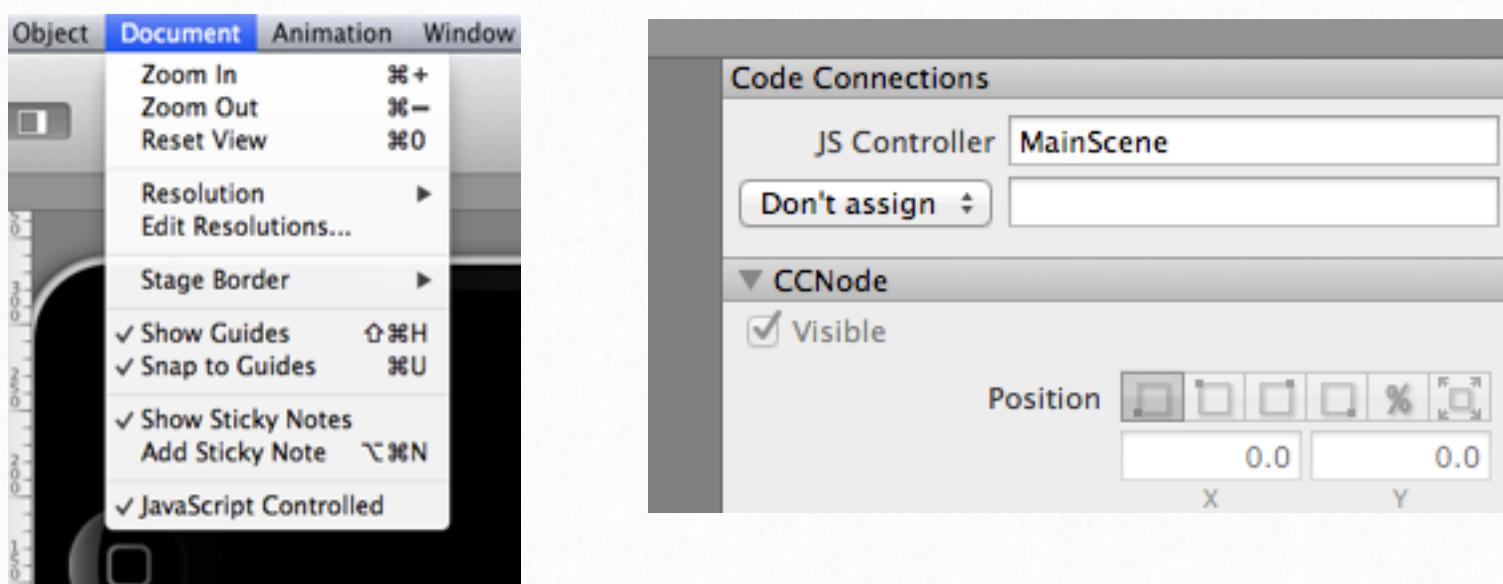
Like we said before that the cocosbuilder support cocos2d-html5 now but this dose not mean that you can use the ccbi file generated by cocosbuilder directly in both cocos2d-iphone and cocos2d-html5 -- that's not gonna work 

It's not as simple as "copy and paste".

Remember that we need to bind our ccbi to a centen scene and that's done by setting the "custom class" in "code connections" section:



As for cocos2d-html5 we need to set the "JS Controller" instead (it can be found in the "code connections" section by checking the "Document->JavaScript Controlled"):



The point is that we can not set the "Custom Class" and "JS Controller" at the same time now which means the ccbi file you generated with the "Custom Class" selected can only be used for cocos2d-iphone and the "JS Controller" only for cocos2d-html5.

And another thing you need to watch out for is cocosbuilder won't warn you for unsaved files so you'd better "save" all the files before publishing them to make sure that you get the ccbi with modification you want or you'll get mad with fact that things not working as you expected



You might notice a lot of warnings while loading ccbi in cocos2d-html5:

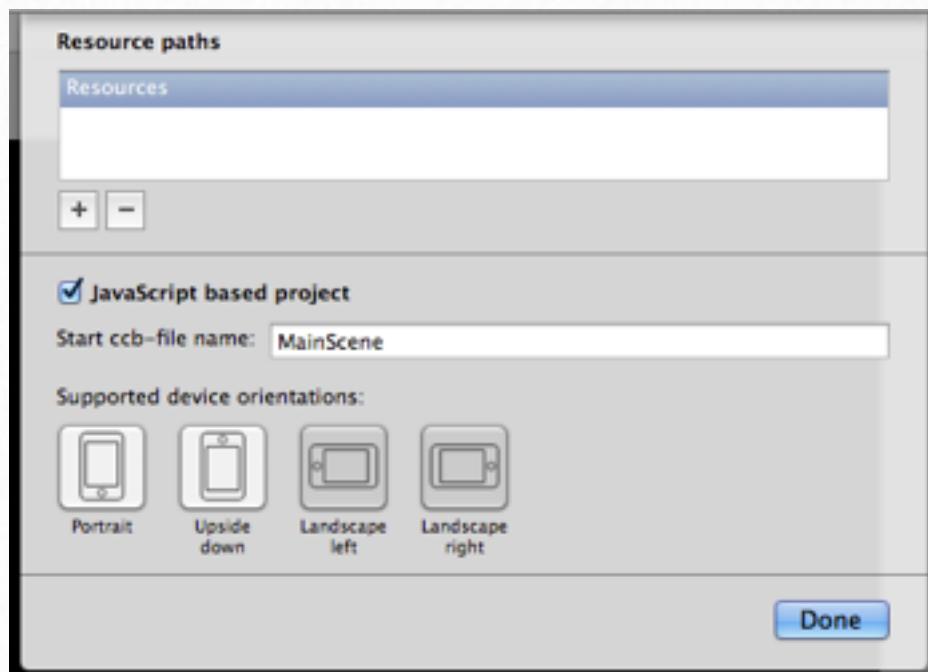
```
Assertion failed: Unexpected property: 'keyboardEnabled'  
cocos2d-html5.min.js:9Assertion failed: Unexpected property: 'touchEnabled'  
cocos2d-html5.min.js:9Assertion failed: Unexpected property: 'touchEnabled'  
cocos2d-html5.min.js:9Assertion failed: Unexpected property: 'accelerometerEnabled'  
cocos2d-html5.min.js:9Assertion failed: Unexpected property: 'accelerometerEnabled'
```

and that's because the CCReader is using "touchEnabled", "accelerometerEnabled" and "keyboardEnabled" while the cocos2d-html5.min.js uses "isTouchEnabled", "isAccelerometerEnabled" and "isKeyboardEnabled". One quick fix here is you can rename all the "isXXX" in cocos2d-html5.min.js to "XXX".

And you might also meet error like below:

```
.NS_ERROR_DOM_BAD_URI: Access to restricted URI denied
```

A possible reason for this: the js file you specified in “Start ccb-file name” (Project Setting) in cocosbuilder dose not exist.



Another one:

.NS_ERROR_ILLEGAL_VALUE: Component returned failure coder

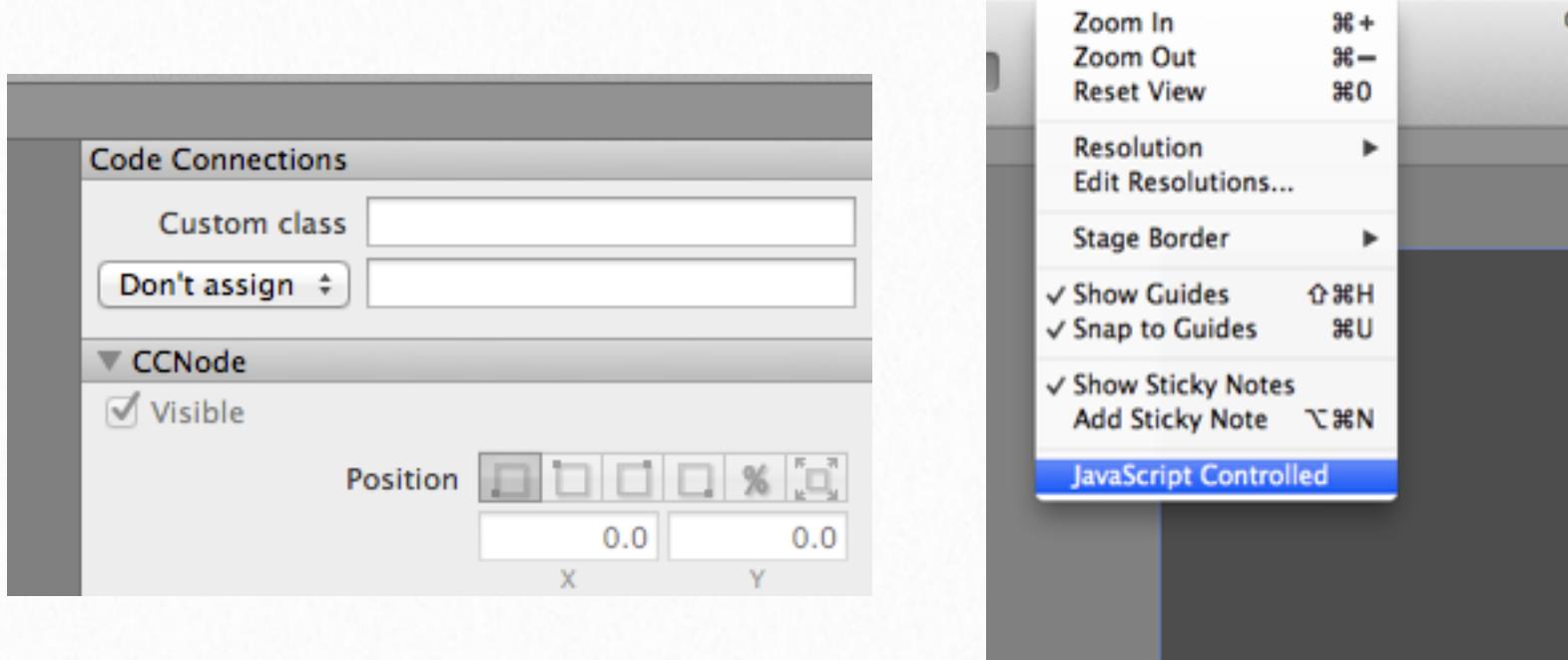
this might caused by bad path for the resource.

And still we might get meaningless errors sometime:

*TypeError: a is null (cocos2d-html5-min.js:xxx)
TypeError: b is null (cocos2d-html5-min.js:xxx)*

it's quite annoying since there is little information from the error and all we can do is look into the code and where the error occurs and find what the “a, b” stands for 😠

For the second error here “b is null” is caused by using the ccbi file generated while not having the “JS Controller” checked (just like mentioned above) -- or you have just set the “JS Controller” but forgot to **SAVE IT**.



The traps I showed you in this section belong to those kind that will drive you nuts and is really hard to figure out why.

And the traps I'm gonna show your in the next chapter are totally different since actually they should not be called traps -- it's just the special features in cocosbuilder+cocos2d-html5 are playing the tricks



7

cocosbuilder in-depth - MVC in cocos2d-html5

Before we start this chapter I need to warn you that this is the hardest part of this book and I also need to confess that I only understand 80% of this chapter myself.

But still I want to show you my understanding and if you find some wrong you can drop me a note and I'll fix it as soon as possible and this not apply to this single chapter but the whole book as well :)

First I'll you a important image of “Class structure” here which you might need to go back

and check when you get confused during this chapter



```
▼ MainMenuScene
  ► aboutButtonLabel: e
    controllerName: "MainMenuScene"
  ► helpButtonLabel: e
  ► highScoreButtonLabel: e
  ▼ rootNode: e
    _accelerometerEnabled: false
  ► _actionManager: e
  ► _anchorPoint: cc.Point
  ► _anchorPointInPoints: cc.Point
  _cacheDirty: true
  ► _children: Array[16]
  ► _contentSize: cc.Size
  _ignoreAnchorPointForPosition: true
  _initializedNode: true
  _inverseDirty: true
  _isMouseEnabled: true
  _orderOfArrival: 0
  ► _parent: e
  ► _position: Object
  _reorderChildDirty: false
  _running: true
  _scaleX: 1
  _scaleY: 1
  ► _scheduler: e
  _super: undefined
  _tag: -1
  _touchEnabled: true
  _touchMode: 0
  _touchPriority: 0
  _transformDirty: true
  _transformGLDirty: true
  _zOrder: 0
  ► animationManager: e
  ► controller: MainMenuScene
  ► getActionManager: function (){return this._actionManager}
  ► getScheduler: function (){return this._scheduler}
  ► onEnterTransitionDidFinish: function () {
  ► __proto__: e
  ► startPlayButtonLabel: e
  ► __proto__: Object
```

Watch out for the dynamic feature in javascript

Before we start the MVC part there is a simple but tricky thing you need to watch out for that will be -- setting properties of an object in cocos2d-html5.

A quick example:

We can use the set function (or modify the property directly) in cocos2d-iphone like this:

cocos2d-iphone:

```
aSprite.position = ccp(50, 50);
[aSprite setPosition:ccp(50, 50)];
bSprite.position = aSprite.position;
```

But remember we should never set the property directly in cocos2d-html5 like this:

```
aSprite.position = cc.p(50, 50);
```

it seem OK -- no error no warning but you'll never want you want -- the sprite won't show at (50, 50). And the reason is the "position" property of CCNode in cocos2d-html5 is actually "_position" -- it's not "position".

And while you try to set the "position" of the sprite which dose not exist javascript will create that property on the fly and set it to (50, 50) but that's apparently not what we want.

even if I write something like this (still no error no warning since it's right) :

```
aSprite.areYourKiddingMe = true
```

This really might confuse someone who uses cocos2d-html5 for the first time.

And one more short example:

```
var winSize = cc.Director.getInstance().getWinSize();
var aSprite = cc.Sprite.create("xxx.png");
aSprite.setPosition(cc.p(winSize.width / 2, winSize.height / 2));
aSprite.setScale(0.5);
aSprite.setColor(cc.black);
this.addChild(aSprite, 1, 1);
```

Just remember using the set/get will keep you just safe 😊

MVC (View) in cocosbuilder + cocos2d-html5

Since it's quite difficult to explain MVC without an example so here we go.

- Adding a sprite to scene:

cocos2d-iphone:

```
// XXXScene
-(void) didLoadFromCCB {
    CCSprite *aSprite = [CCSprite initWithFile:@"xxx.png"];
    [self addChild:aSprite];
}
```

cocos2d-html5:

```
XXXScene.prototype.onDidLoadFromCCB = function () {
    var aSprite = cc.Sprite.create("xxx.png");
    this.addChild(aSprite);
};
```

The code in cocos2d-iphone works fine as expected but the one in the coco2d-html5 won't, it will give you an error saying "function undefined: addChild" 

The reason why we get this error is because cocosbuilder uses the MVC mode in cocos2d-html5:

You can consider the scene we are binding to the ccbi in cocos2d-iphone as the "View" in MVC so adding a sprite with "[self addChild:aSprite]" is correct.

But, it's not the same when it comes to the cocos2d-html5, the scene we are binding to the ccbi is actually the “Controller” so adding a sprite with “`this.addChild(aSprite);`” is absolute wrong.

But where on earth is the “View” in cocos2d-html5, we can try to look into the code:

```
cc.BuilderReader.load = function (ccbFilePath, owner, parentSize, ccbRootPath) {  
    //.....  
    // Create a document controller  
    var controller = new _ccbGlobalContext[documentControllerName]();  
    controller.controllerName = documentControllerName;  
    innerNode.controller = controller;  
    controller.rootNode = innerNode;  
    //.....  
}
```

Yep, that's the “`rootNode`” property -- that's the “View” we're looking for -- so the solution is quite obviously

```
this.rootNode.addChild(aSprite);
```

One more example here - binding a selector to the scene:

```
MainScene.prototype.onOptionPressed = function () {}
```

as you can see the function “`onOptionPressed`” is bound to the “Controller” not the “View”

Let's prove it with the touch function:

```
MainScene.prototype.onTouchesBegan = function (touches, event) {  
    cc.log("MainScene.prototype.onTouchesBegan");  
};
```

You will not get any message after running and touching the screen, why, well it's obviously that the touch is applied to the "View" not the "Controller" of the scene so adding the touch handler to the controller won't do the trick.

Let's try to bind the function to the view:

```
XXXScene.prototype.onDidLoadFromCCB = function () {
    this.rootNode.onTouchesBegan = function(touches, event) {
        cc.log("this.rootNode.onTouchesBegan");
    };
};
```

And this time we can see the "this.rootNode.onTouchesBegan" is printed on the console while touching the screen.

So you should always make it clear that if you need the "View" or the "Controller" while using cocosbuilder with cocos2d-html5



MVC (Controller) in cocosbuilder + cocos2d-html5

It will be easier for us to understand the “Controller” after we have learnt the “View” in cocos2d-html5.

The scene we are binding to the ccbi in cocos2d-html5 is actually the “Controller” and if we want to call function from inside the “onTouchesBegan”:

```
XXXScene.prototype.onDidLoadFromCCB = function () {
    this.rootNode.onTouchesBegan = function(touches, event) {
        cc.log("this.rootNode.onTouchesBegan");
        this.handleTouch();
    };
};

XXXScene.prototype.handleTouch = function () {}
```

but this won't work 😞

And this boils down to a very confusing conception in javascript -- the keyword “this” and what dose “this” stands for in the context.

In C++, the “this” keyword is a pointer to the current instance of the class in a function in which the “this” is used.

But, in javascript it's quite different, it's way above my ability to explain this here but one thing I know is “**this stands for the object calling the function**”.

And know let's go back to the example above and focus on the “this.handleTouch();” which is obvious wrong since the caller of the “onTouchesBegan” is the “View” not the “Controller” so the “this” keyword inside function is the “View” and the “handleTouch” function is binded to the “Controller”.

So let's check the code in CCBReader again:

```
cc.BuilderReader.load = function (ccbFilePath, owner, parentSize, ccbRootPath) {  
    //.....  
    innerNode.controller = controller;  
    controller.rootNode = innerNode;  
    //.....  
}
```

And we can find the solution here:

```
XXXScene.prototype.onDidLoadFromCCB = function () {  
    this.rootNode.onTouchesBegan = function(touches, event) {  
        cc.log("this.rootNode.onTouchesBegan");  
        this.controller.handleTouch();  
    };  
};
```

OK, finally a long example for you to check if you actually understand this:

```
var MainScene = function() {  
    cc.log("MainScene");  
};  
MainScene.prototype.onDidLoadFromCCB = function () {  
    cc.log("MainScene.prototype.onDidLoadFromCCB");  
    this.rootNode.setTouchEnabled(true);  
    this.test();  
    this.rootNode.onTouchesBegan = function( touches, event) {  
        cc.log("this.rootNode.onTouchBegan");  
        this.controller.test();  
    };  
  
    var testScene = cc.BuilderReader.load("TestScene.ccbi");  
    testScene.controller.delegate = this;  
    testScene.controller.test();  
};
```

```
MainScene.prototype.test = function () {
    cc.log("MainScene.prototype.test");
};

MainScene.prototype.onTouchesBegan = function( touches, event) {
    cc.log("MainScene.prototype.onTouchBegan");
    return true;
};
```

```
var TestScene = function() {
    cc.log("TestScene");
    this.delegate = null;
};

TestScene.prototype.onDidLoadFromCCB = function () {
    cc.log("TestScene.prototype.onDidLoadFromCCB");
};

TestScene.prototype.test = function () {
    cc.log("TestScene.prototype.test");
    this.delegate.test();
};
```

The output after running the app:

```
MainScene
MainScene.prototype.onDidLoadFromCCB
MainScene.prototype.test
TestScene
TestScene.prototype.onDidLoadFromCCB
TestScene.prototype.test
MainScene.prototype.test
```

and after touch:

```
this.rootNode.onTouchBegan  
MainScene.prototype.test
```

Now it's time to go back to explain "that":

```
var that = this;  
setTimeout(function(){that.someSelector();}, 1.0 * 1000);
```

we know that the keywords "this" in javascript keeps changing what it means so we store it in a variable called "that" when we are sure it's pointed to what we want at that point.

Finally, the most difficult part is over. I know it might be quite confusing especially for those who are not familiar with javascript so best the way to understand this is to create a project and try everything yourself.

Now we can finally port our game from cocos2d-iphone to cocos2d-html5 under the help of cocosbuilder 

We'll talk about something interesting next chapter and now it's time to grab our coffee cup again 

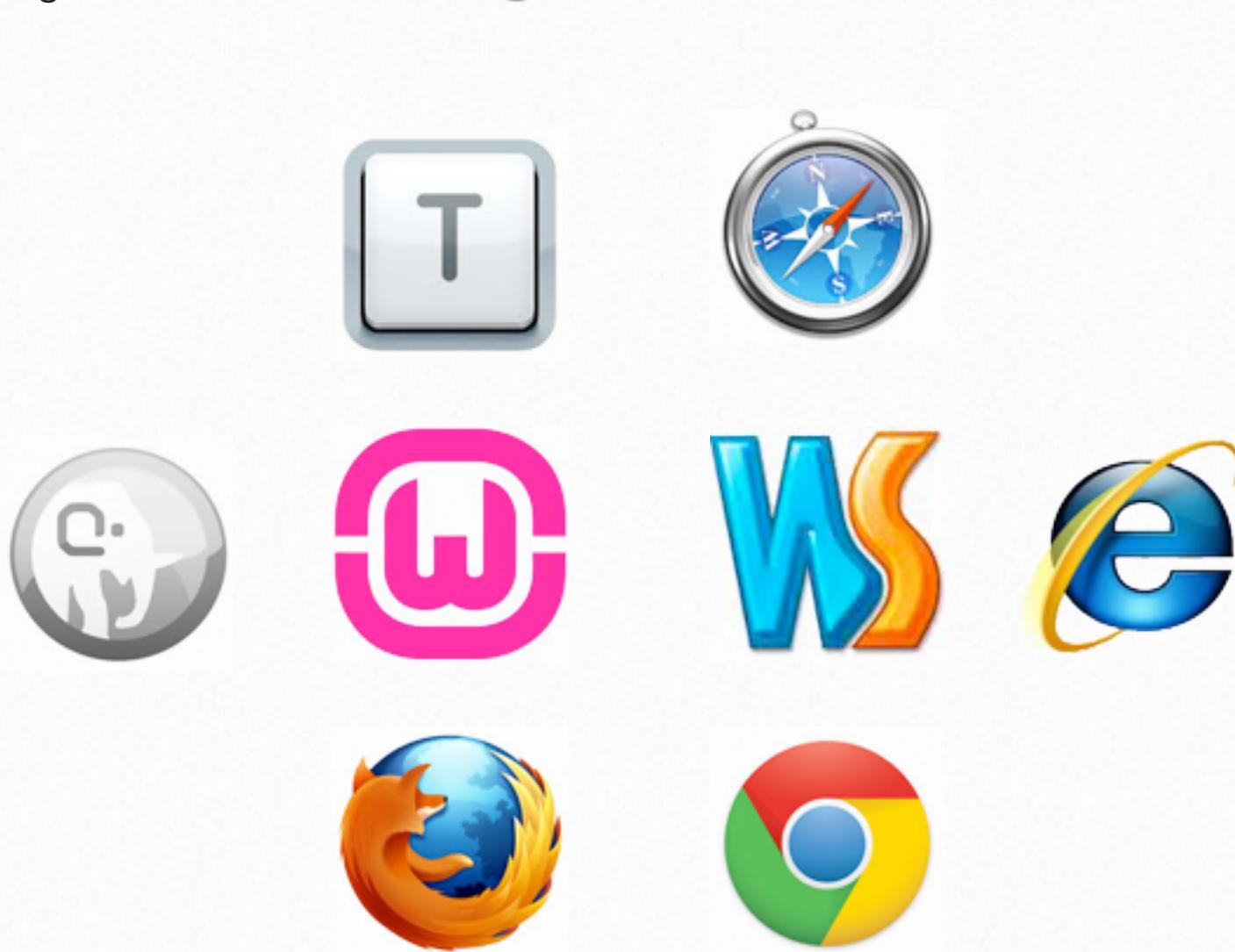
8

Play with cocos2d-html5 - Various IDE and debug tools

A good IDE means a lot to the developer since it can give you cool features such as code highlight, autocomplete and with the IDE you can debug your game in a convenient way.

Besides choosing a good IDE for formal development there're also some other ways to debug for our html5 game.

OK, let's go and meet our friends 🕸️

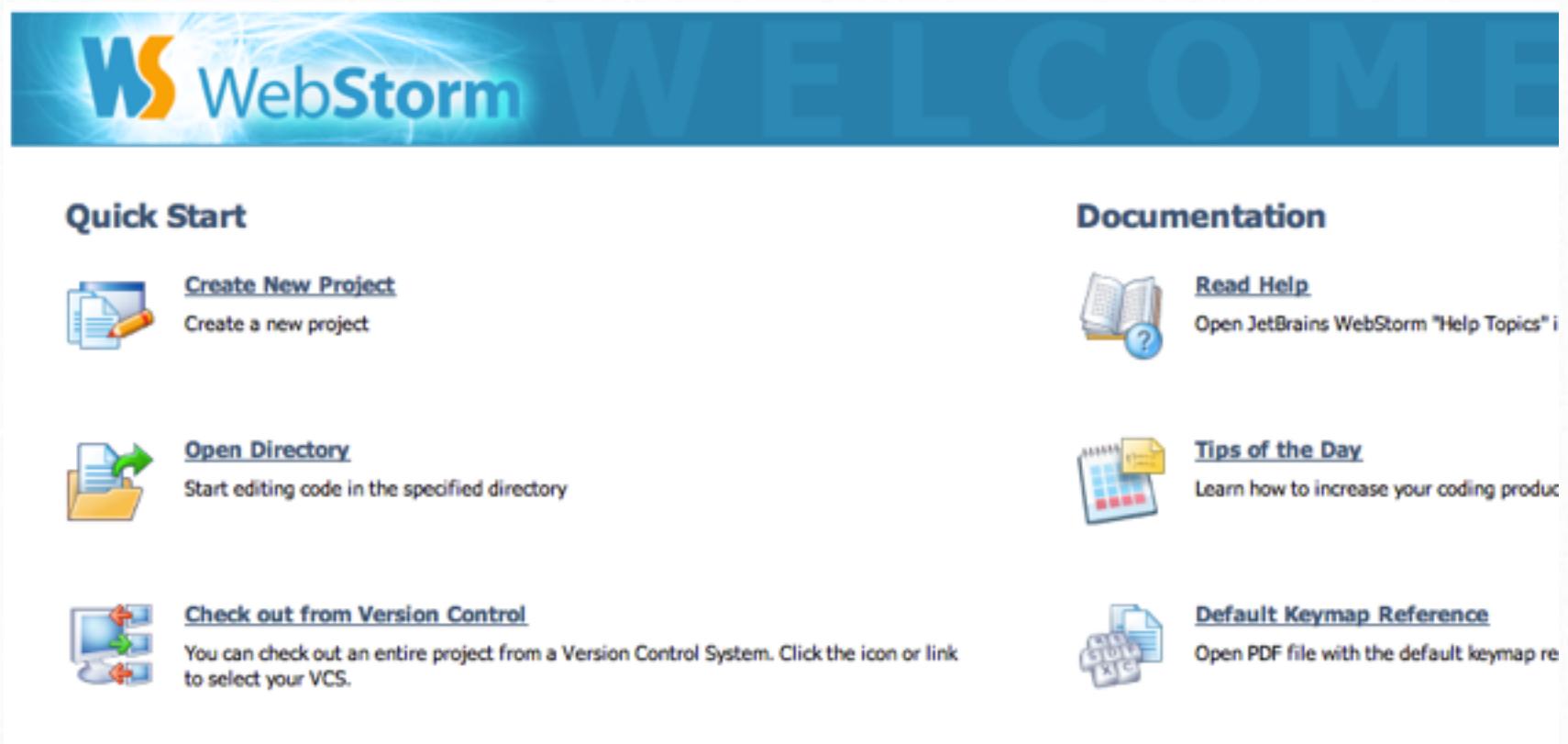


IDE for development - Jetbrains WebStorm

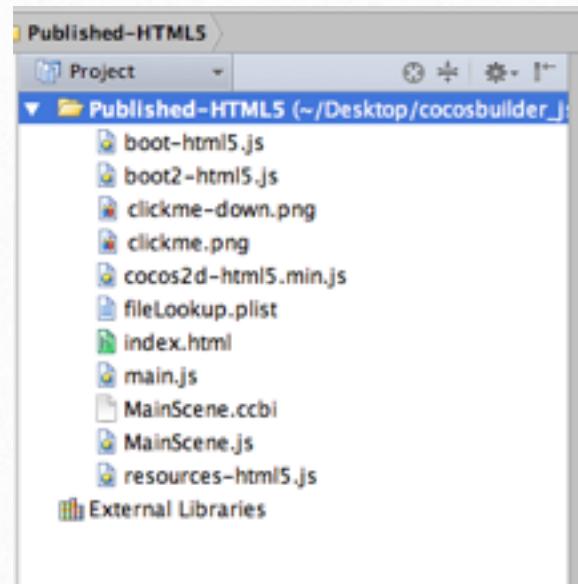
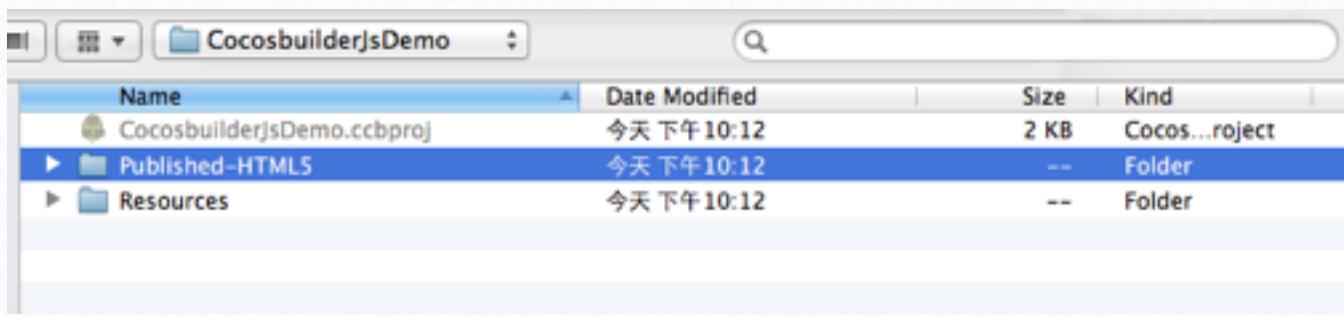
Jetbrains WebStorm is the most intelligent JavaScript IDE: JavaScript + CSS + HTML Editor with refactorings, code completion and on-the-fly code analysis.

And here is a example of how to import our game using cocosbuilder in to the WebStorm.

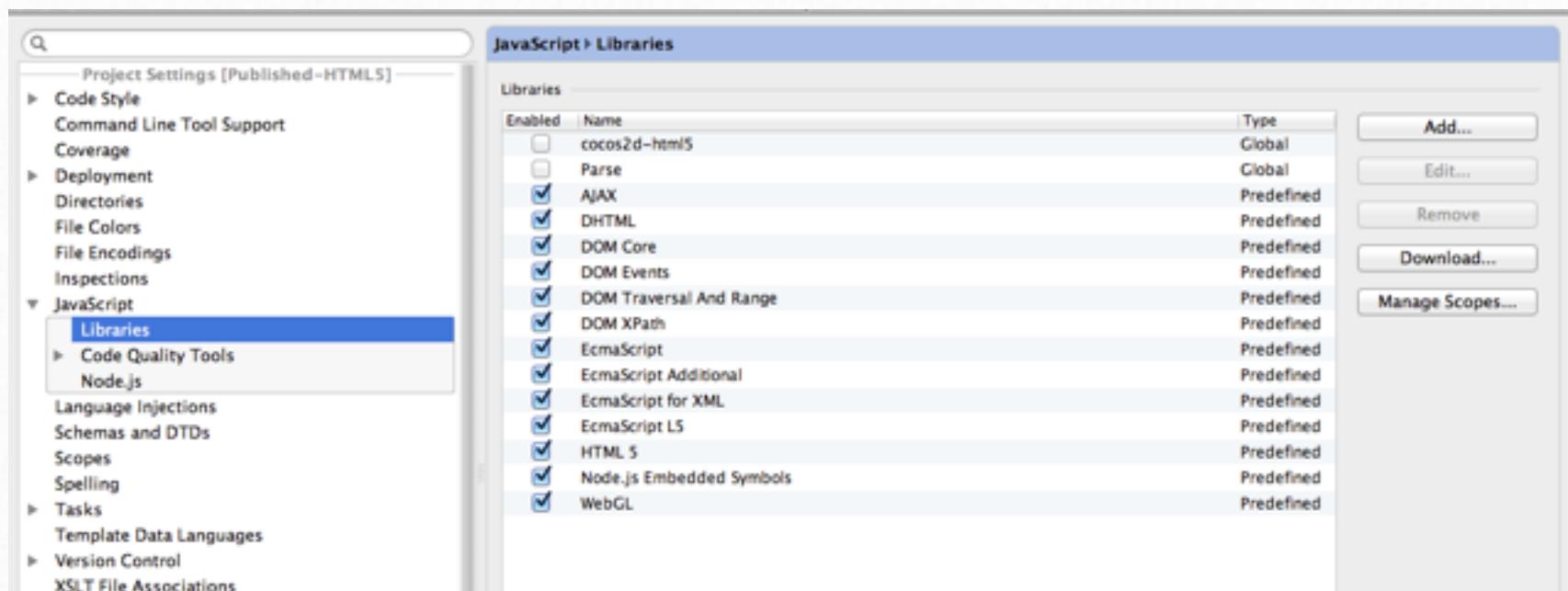
- First go and download the software
- Then click the “Create New Project”



- Choose the folder published by cocosbuilder



- Import the cocos2d-html library (if you're using the cocos2d-html5-min.js then you don't need to do this)



- And that's all and now we can give it run.

```
<!DOCTYPE HTML>
<html>
<head>
    <meta charset="utf-8">
    <title>CocosBuilder Generated HTML5</title>
</head>
<body style="text-align: center; background: #f2f6f8;">
    <div style="display:inline-block; width:auto; margin: 0 auto; background: black; position: relative; height: 320px; border-radius: 50%; overflow: hidden; width: 480px;">
        <canvas id="gameCanvas" width="480" height="320"></canvas>
    </div>
    <script src="boot-html5.js"></script>
</body>
</html>
```

Run

0. Edit configurations...

2. index.html ▾

Hold ⌘ to Debug

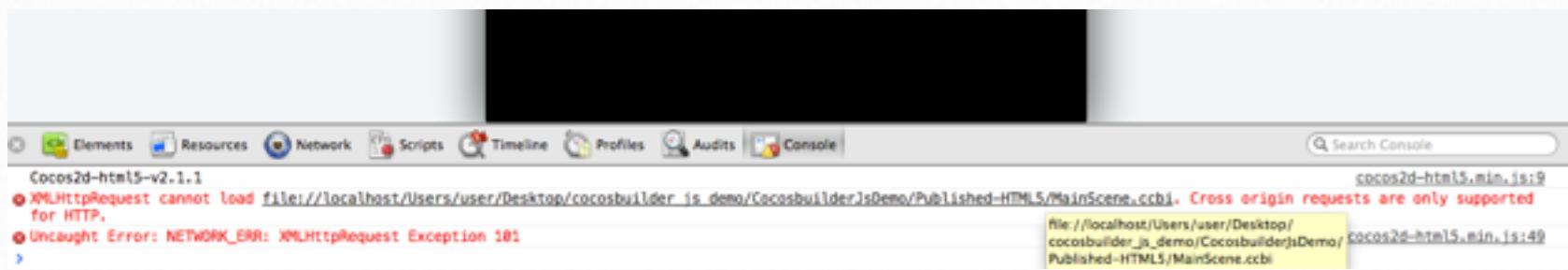
Actions

Debug

Edit...

Save temp configuration

but unfortunately we might get following error:



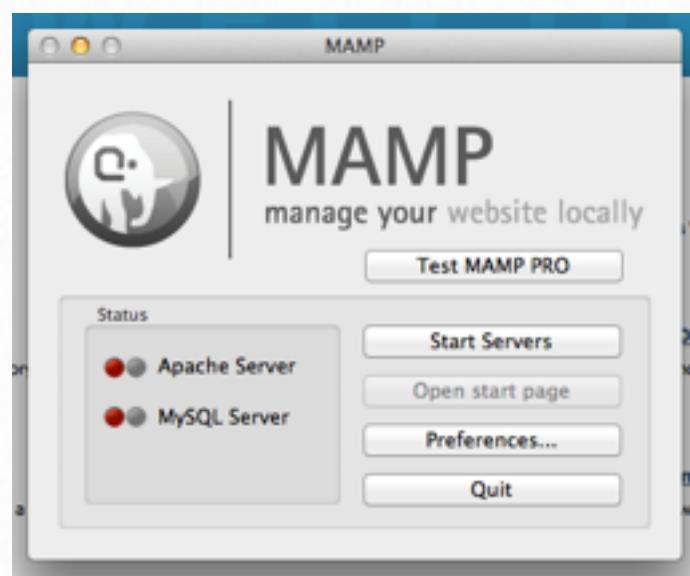
the reason we are getting this error is because the browser we are using (google chrome) dose support “cross origin request” -- so we can either change out browser to a new one that supports “cross origin request” or we can just set up a local server to run our game in which is better 😊

Setting up a local server

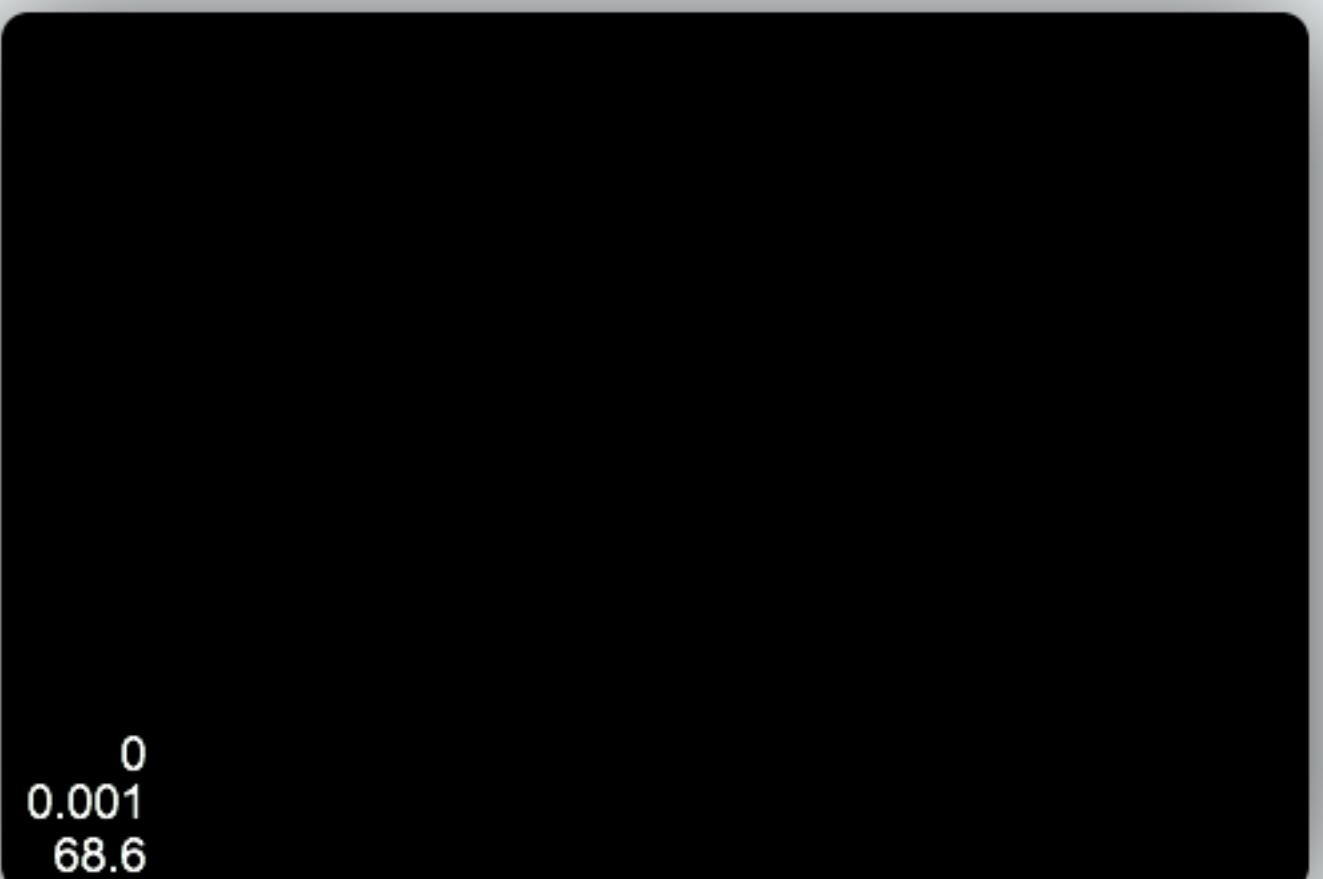
Now we know that we need to set up a local server for our game so let's get started (you can use firefox which supports cross origin request but I strongly recommend you set up a local server).

We can use the WAMPServer for Windows OS and MAMP for MacOS to set up the local server and it's quite easy:

- First go and download the software
- And after installing copy the folder published by the cocosbuilder to the "www" folder under Windows OS ("htdocs" folder under Mac OS)
- Start service



And now we can run our game without getting the error we see before.



0
0.001
68.6

Co-debug our game in LAN

There is another benefit after setting up the local service -- all the device in the LAN can access your server and co-debug the game together.

And it's quite easy -- all we need to do is give privilege to the device in the same LAN to access our server which is not allowed by default.

Let's take WampServer as example:

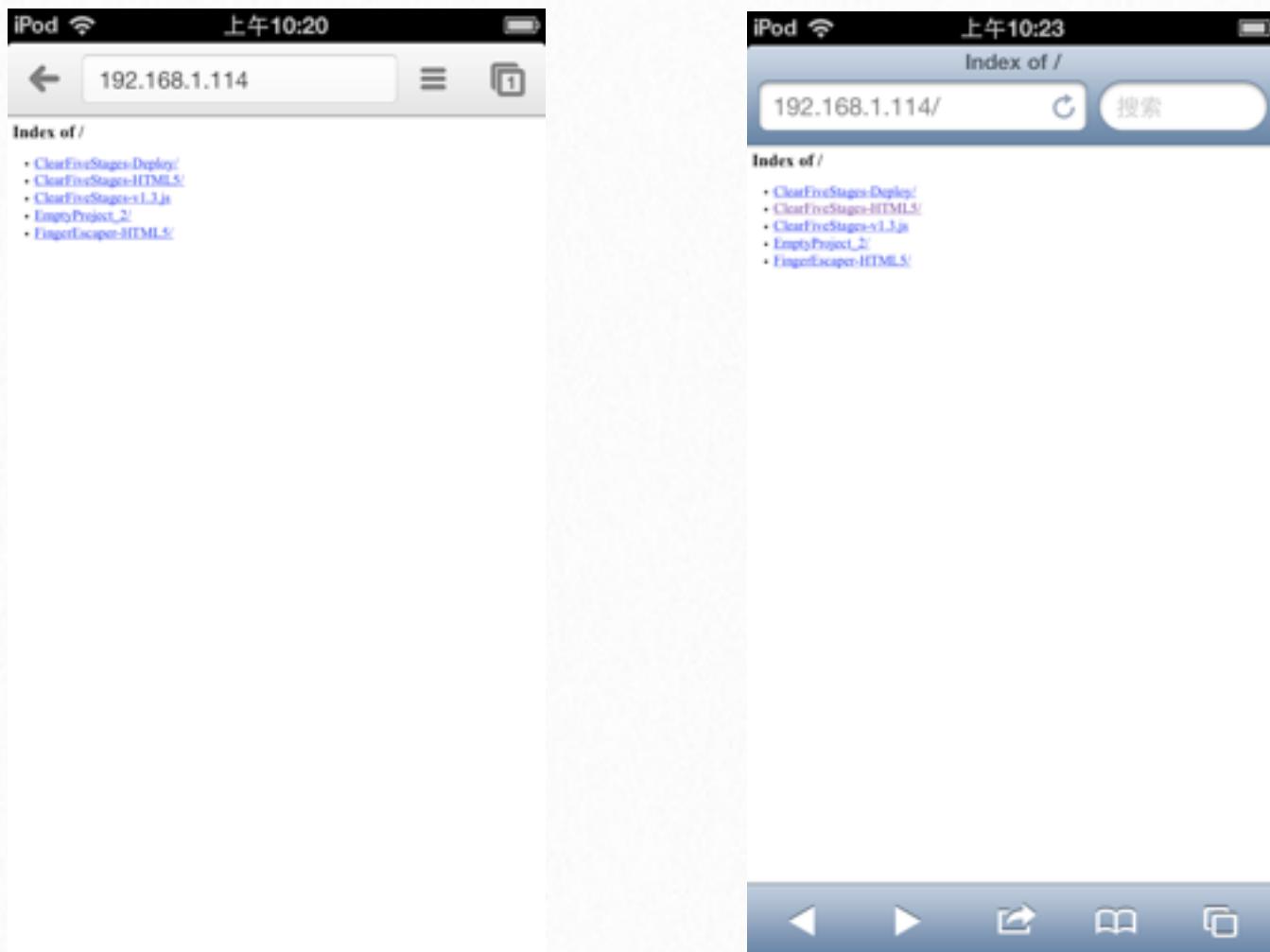
- Open the httpd.conf file



- Edit the content comment out the “Deny from XXXX” and then add “Allow from all”

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Order deny,allow
    #Deny from all|
    Allow from all
    Satisfy all
</Directory>
```

And now we can access the game on the server from the device in the same LAN.



One more thing to notice here -- if you're using browsers on PC to access the game then you can use the built-in debug tool to debug the game but if you wan't to use the mobile device whose browsers do not have debug tools you need something else to help you.

And that is -- “firebug”. Firebug is a tool for web development. Allows inspect, edit and monitor CSS, HTML, JavaScript and Net requests in any web page.

We can use a js file they provided called “firebuglite.js” to add console to our browser in the mobile device.

All you need is add this line in the index.html:

```
<script type="text/javascript"
src="https://getfirebug.com/firebug-lite-debug.js">
</script>
```

and now after loading the game you can see a console pops up:

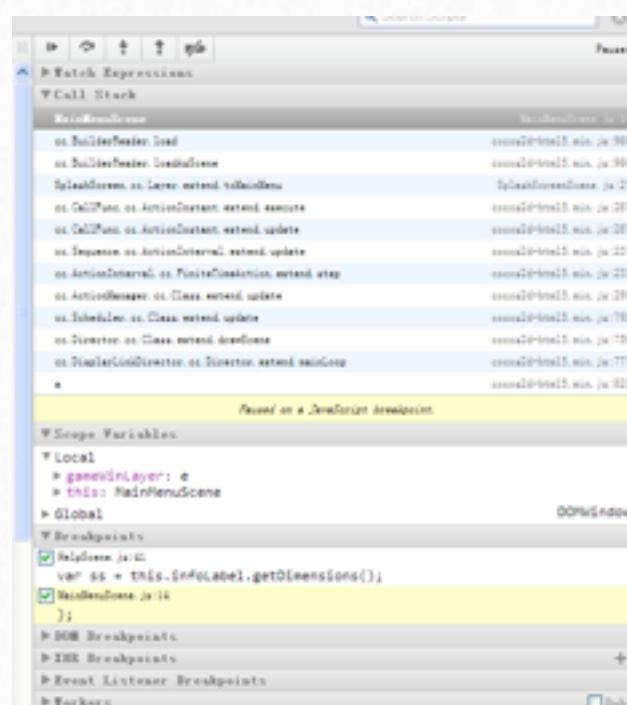
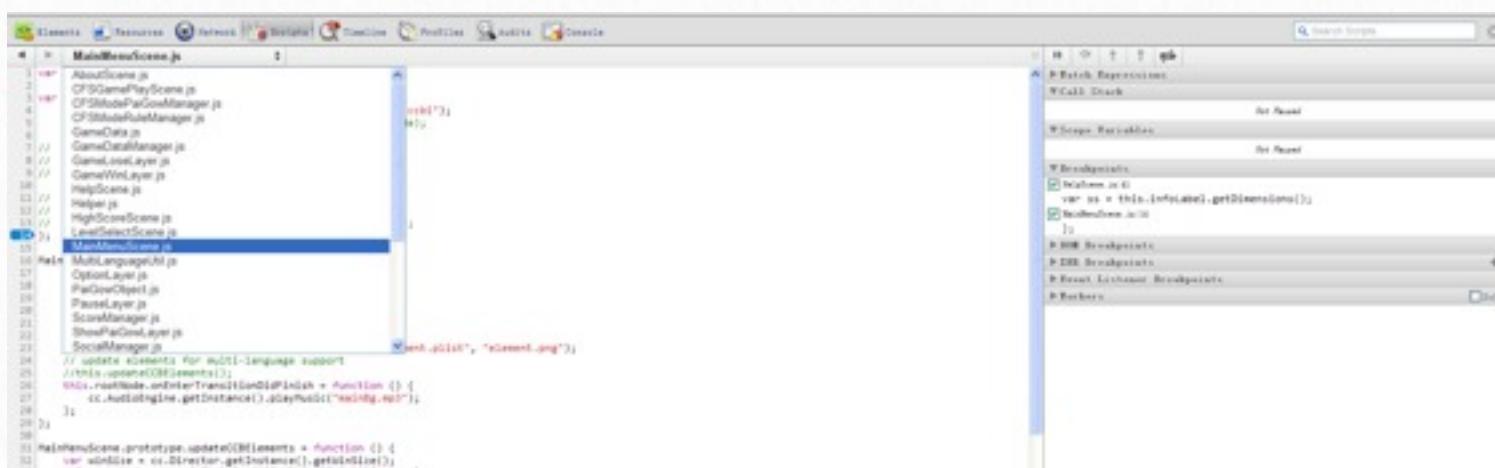


Cool isn't it  You can see all the log output and other functions are supported as well.

Quick debug - notepad + Google chrome

When you don't have any IDE at hands and still want to modify & debug your game then the notepad + Google chrome is one good choice.

The built-in debug tool of Google chrome is quite powerful you can see the log output in the console and even set breakpoint in the script to help you with debug.



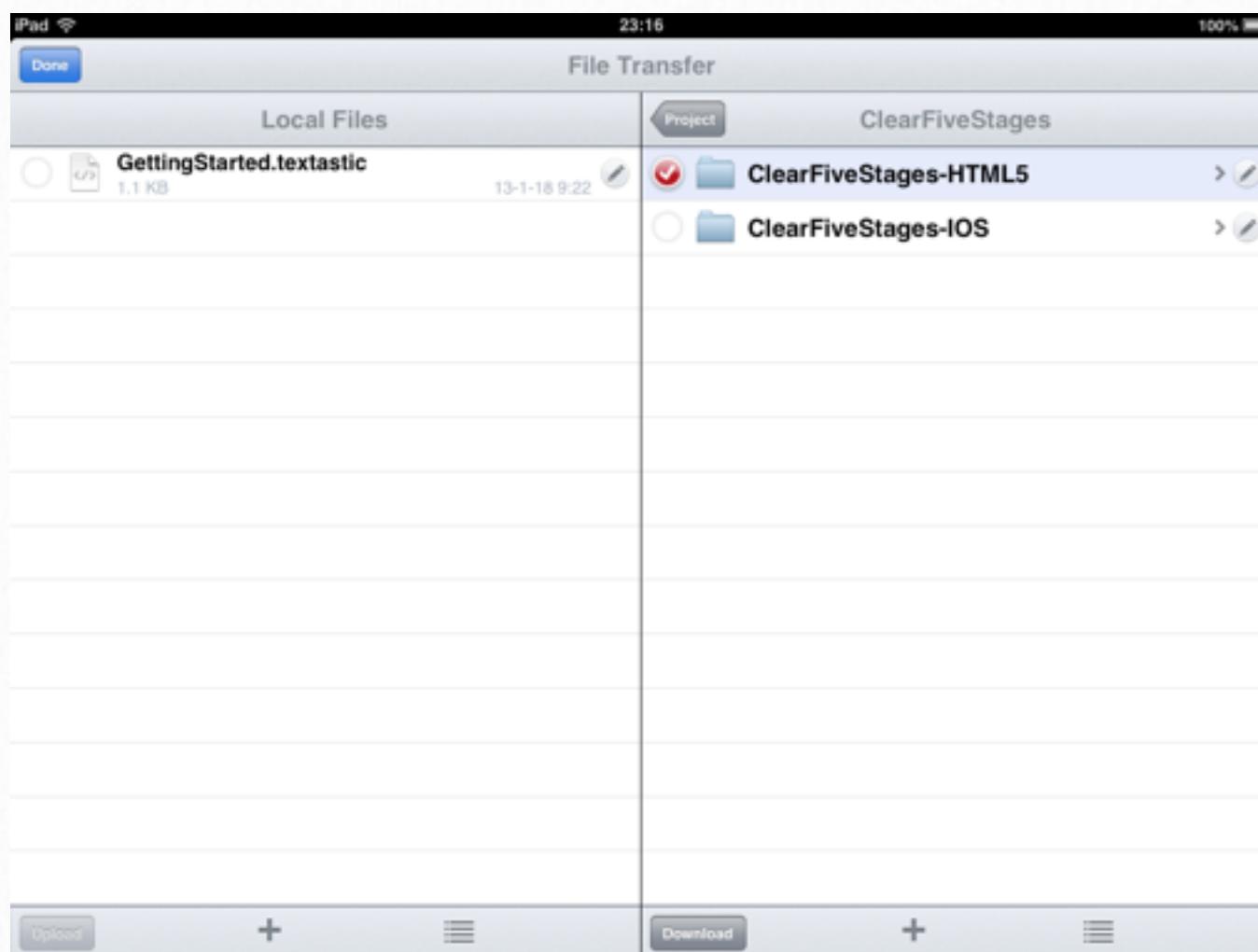
And here's some tips I would like to share:

- You can use the check your code files in the drop down list in the “script” tab and if you found some files missing then it means that there is format errors in your code (missing ‘;’ or ‘}’)
- You can set breakpoint, set in/step out/step over in the “script” tab
- You can find the output log in the “console” tab and you can click the red arrow to expand the “call stack” on an error message which can help you debug
- Something we will find the modification we made to the code dose not take effect when running the game, then you can check the file you have just modified by selecting it in the drop down list and check if the file loaded right now is the one you just modified and if not then “Clear browser cache” will help you to solve this problem

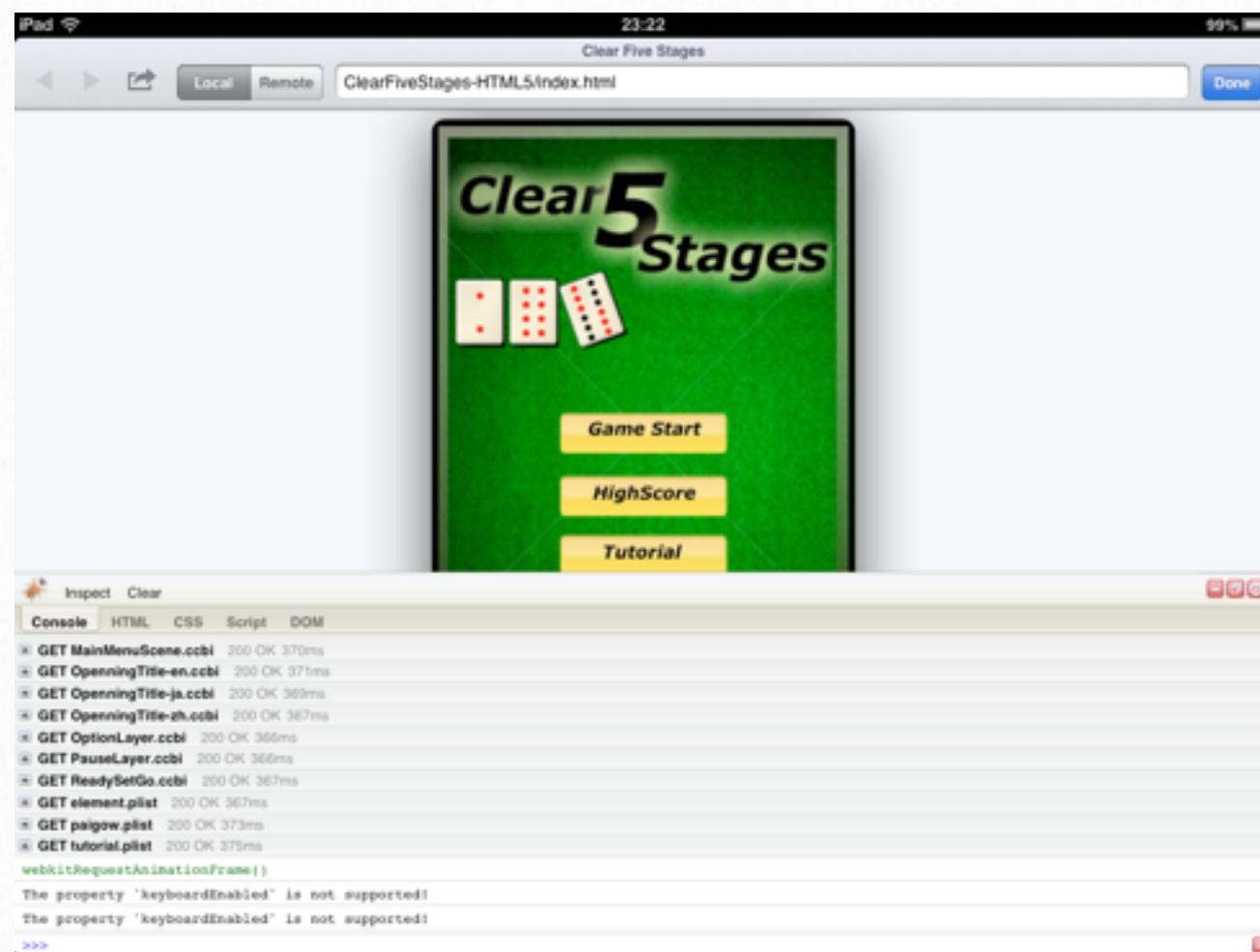
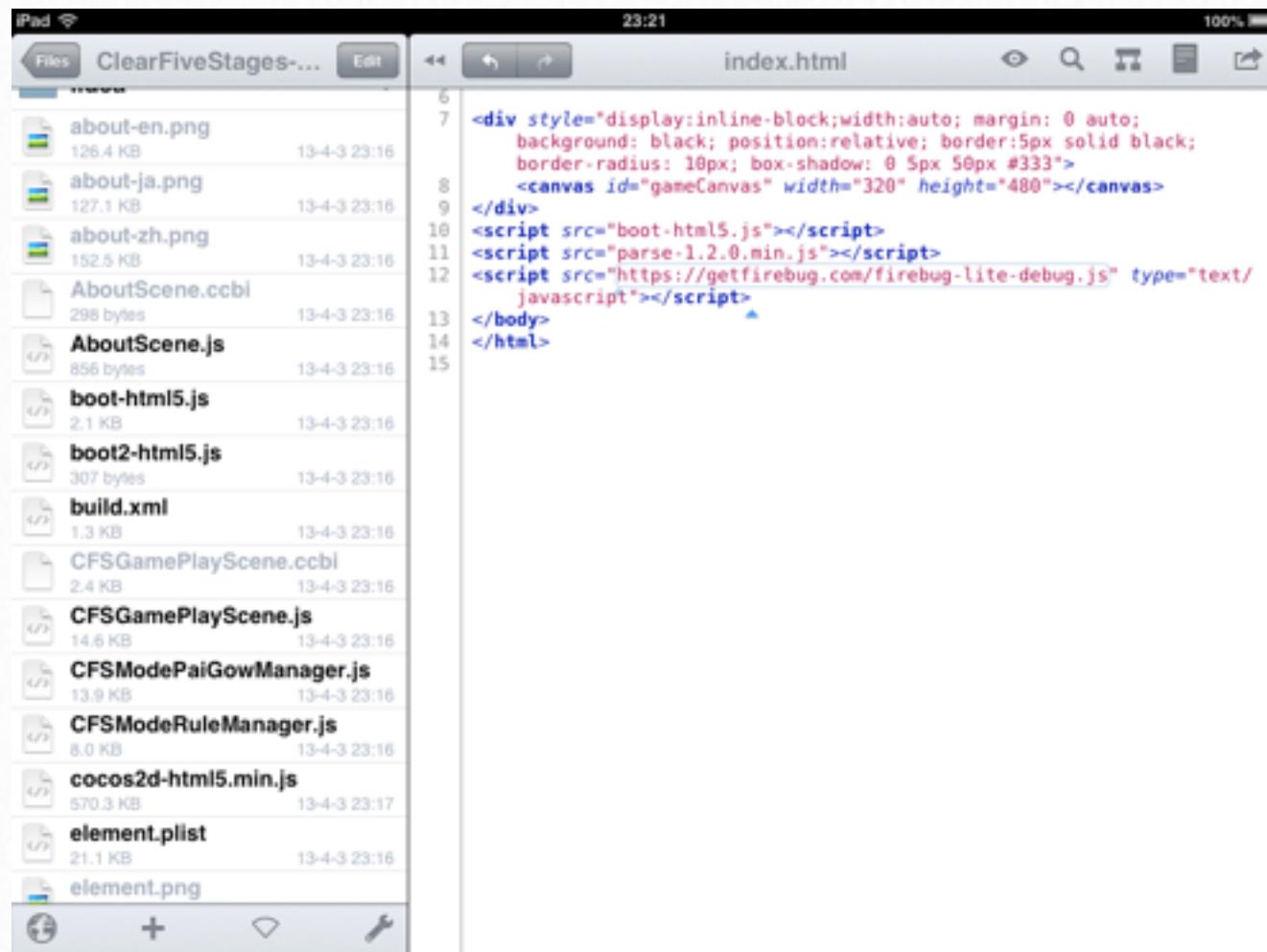
Debug offline on ipad - textastic + firebug

The last we gonna talk about is how to edit && debug on ipad even when offline under the help of an app named -- textastic.

Since textastic can connect to our Dropbox, we can put our game code on dropbox and then clone it to textastic edit && debug and then sync it back to dropbox while we have network connection.



and we can also use the “firebuglite.js” to help debug on ipad:



Now we have setup our test and debug environment don't forget to test the game carefully since we'll pack them for release in the next chapter



9

Time for release - Pack your game with ant

After testing out game for a while make sure that most of the bugs are gone and we can start to prepare for releasing our game.

We need to pack our game use ant to minimize our code so that our game will load quickly on the internet.



Using ant to pack your game

It's quite easy to pack your game with ant. First we need to install "ant" if we don't have it and then we need a "build.xml" file in our game folder, you can copy one from inside the cocos2d-html5 source code.cocos2d-html and don't forget the "compiler.jar".

The file is pretty straightforward, it looks like this:

```
<?xml version="1.0"?>
<project name="Javascript compress project" basedir=". default="co-
mpile">

    <taskdef name="jscomp"
classname="com.google.javascript.jscomp.ant.CompileTask"
        classpath="${basedir}/compiler.jar"/>
    <target name="compile">
        <jscomp compilationLevel="simple" warning="quiet"
            debug="false" output="${basedir}/..xxxxxx-release.js">
            <sources dir="${basedir}">
                <file name="AboutScene.js" />
                <file name="CFSGamePlayScene.js" />
                .....
            </sources>
        </jscomp>
    </target>
</project>
```

All you need is modify the path and filename according to your own game.

After that go and run "ant" command in your folder and you'll get a single file called "xxxxxx-release.js".

Code modification

When we have our “xxxxxx-release.js” packed then we need to modify our “boot-html5.js” a bit:

```
appFiles:[
  //...
  'xxxxxx-release.js'
  //...
]
```

we should add out new js file while deleting the ones that won’t be used anymore.

You can also pack the cocos2d-html5 source file along with your game code into a single file as well then all you need is to set the “SingleEngineFile” :

```
SingleEngineFile:'xxxxxx-release.js',
appFiles:[]
```

After modify the “boot-html5.js” file you can upload the game to your own site or simply to dropbox to test your game there.

10

Play with the world - Using Parse for Leaderboard in cocos2d-html5

Adding leaderboard function to our game is very important. We have gamecenter to take care of the leaderboard function in our game in cocos2d-iphone but what for cocos2d-html5?

You can implement your own leaderboard server of course but that's too complex and not ness sometimes when all you need is a simple ranking system.

And I'll show you how to get this done under the help of "Parse" -- <http://www.Parse.com> -- in this chapter.

Introduction to Parse

Parse is the cloud app platform for iOS, Android, JavaScript, Windows 8, Windows Phone 8, and OS X.

With Parse, you can add a scalable and powerful backend in minutes and launch a full-featured app in record time without ever worrying about server management. We offer push notifications, social integration, data storage, and the ability to add rich custom logic to your app's backend with Cloud Code.



They have very detailed document and step by step tutorials for each language and if you have got problems using Parse you can ask question in the forum they're willing to help all the time :)

Getting start with Parse

First we need a account of course and then we can create an app for testing.

After creating an app you can find the ID/KEY info that we need to use in our code:

The screenshot shows the Parse.com dashboard for an app named "RaccoonTest". The left sidebar has a blue arrow pointing to the "Application keys" section, which is currently selected. The main content area is titled "Application Keys" and lists several key types with their values redacted in red boxes and "Copy" buttons:

Key Type	Value (Redacted)	Action
Application ID	[REDACTED]	Copy
Client Key	[REDACTED]	Copy
Javascript Key	[REDACTED]	Copy
Windows Key	[REDACTED]	Copy
REST API Key	[REDACTED]	Copy
Master Key	[REDACTED]	Copy

We are using javascript for our game so what we need here is the "javascript Key".

Initialize Parse is easy, first import the parse js file in index.html:

```
<script src="parse-1.2.0.min.js"></script>
```

and then we can initialize Parse in our code like this:

```
Parse.initialize("application id", "javascript key");
```

and that's it



Using Parse to load/save data

The first thing to implement a leaderboard is to load/save data for our game. It is extremely easy to do it with the API provided by Parse:

```
var GameScore = Parse.Object.extend("GameScore");
var gameScore = new GameScore();
gameScore.set("score", 1337);
gameScore.set("playerName", "Sean Plott");
gameScore.save(null, {success: function(gameScore){}, error: function(gameScore, error) {}});
```

The one thing that you should bare in mind is that all the object in Parse is a subclass of Parse.Object.

As you can see to save data all we need is:

- Create a GameObject by extending the Parse.Object
- Then we create a instance of our GameObject
- Then we bind the property we want to the GameObject using “set” function
- And at last all we need is to call the “Save” function to push our game data to Parse

Then let's check how to load data in Parse:

```
var GameScore = Parse.Object.extend("GameScore");
var query = new Parse.Query(GameScore);
query.get("objectID", {success: function(gameScore) {}, error: function(object, error) {}});
```

This shows how to get data from Parse by specifying the objectId property which is automatically generated by Parse while saving the data.

When we have our gameScore object we can get the property using the “get” function:

```
var score = gameScore.get("score");
var playerName = gameScore.get("playerName");
```

and there are 3 special properties created by Parse while saving the data, you can get them like this:

```
var objectId = gameScore.id;
var updatedAt = gameScore.updatedAt;
var createdAt = gameScore.createdAt;
```

On Extending Parse.Object

Sometimes we might want to extend Parse.Object for our own game data object and add some functions to it as well.

Here is an example showing you how to do it:

```
var ScoreData = Parse.Object.extend("ScoreData",
{
  initialize:function () {
    this.playerName = "ANONYMOUS";
    this.playTimeInSec = 99999;
    this.updateScore();
  },
  updateScore:function () {
    this.set("playerName", this.playerName);
    this.set("playTimeInSec", this.playTimeInSec);
  },
  saveScore:function (callback) {
    // .....
  }
});
```

One thing to notice here is “The Parse JavaScript SDK is based on the popular Backbone.js framework.” so here we use the “initialize” function instead of the “init” or “ctor” function.

On Querying and Sorting data

After learnt how to load/save data with Parse it's to learn how to query and sort our data.

An example of how to get the top10 score data:

```
var query = new Parse.Query(ScoreData);
query ascending("playTimeInSec");
query limit(10);
query.find({
  success: function(results) {
    var array = new Array();
    for (var i = 0; i < results.length; i++) {
      var scoreData = results[i];
      alert(scoreData);
    }
  },
  error: function(error) {
    alert("error");
  }
});
```

Again you can see the API is really straightforward, by calling “find” Parse will return the array of data fitting the query condition and then we can use the data for our display 😁



Using anonymous function with asynchronous Parse API

Most javascript functions of Parse will act asynchronously. And we can use “anonymous function” in javascript to work with the Parse API:

For example, saving data to Parse might take a while and we can do it like this:

```
saveScore:function (callback) {
  this.save(null, {
    success: function(scoreData) {
      callback(scoreData);
    },
    error: function(error) {
      callback(null);
    }
  });
}
```

```
ScoreManager.saveScoreData = function (scoreData, callback) {
  scoreData.saveScore(callback);
};
```

and we can pass a “anonymous function” to our saveScoreData function which will be called when the save process is done:

```
ScoreManager.saveScoreData(that.cfsScoreData, function (scoreData) {
  if (!scoreData || !scoreData.id)
    alert("error saving data");
  else
    alert("Data here" + scoreData);
});
```

There are still lots of cool stuffs in Parse which you can take advantage of and implement something cool in your own game, just go and check it



11

Welcome twitter - Using web intents to tweet in game

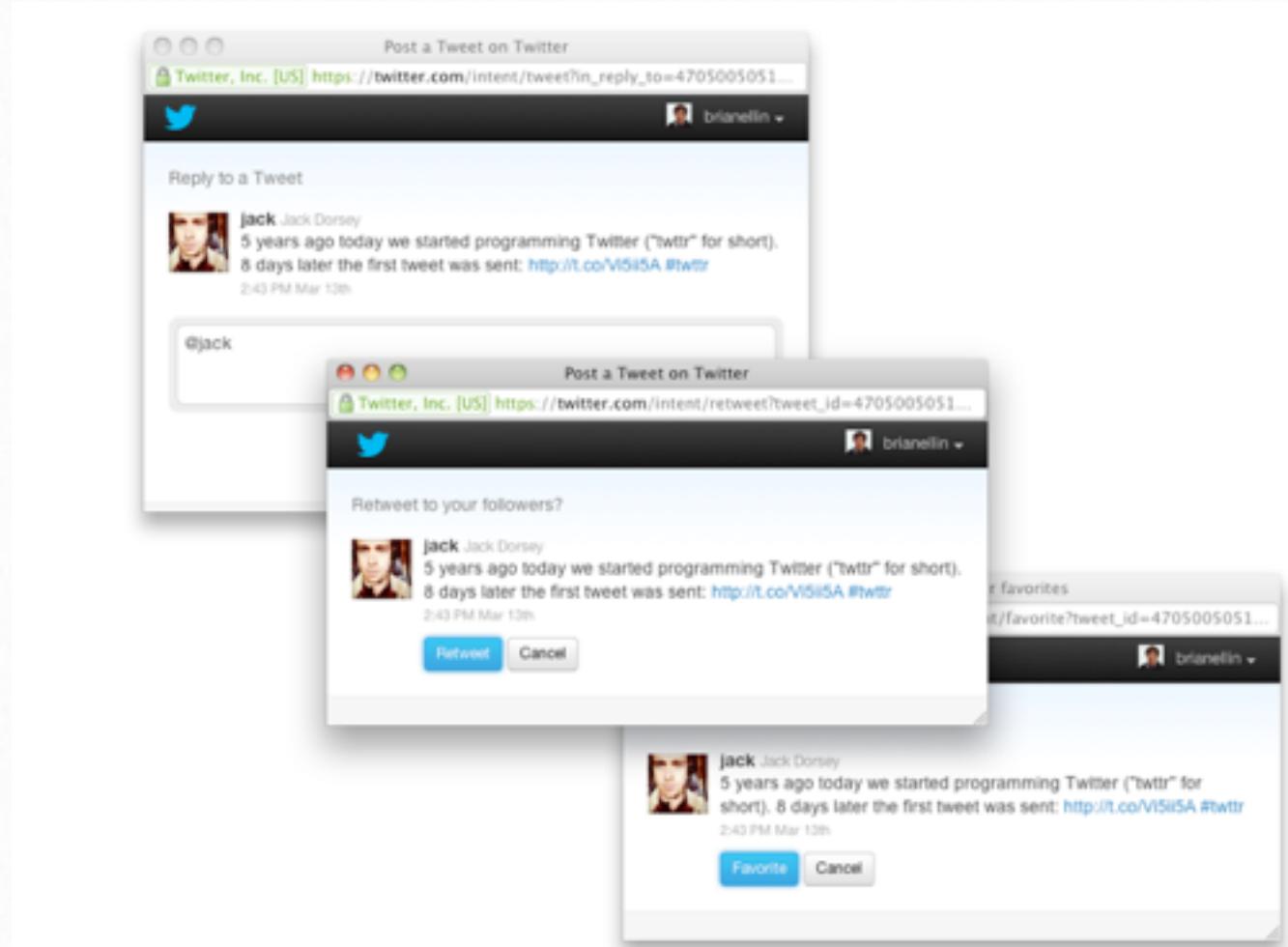
Adding social element to our game is very important in modern games. You can even consider it as a free way of promotion. Twitter is one of the famous social network and the one you should add to your game as well.

It's so easy to integrate twitter into our cocos2d-iphone game since it is integrated in ios 5.0 already and it is also easy to do the same thing in cocos2d-html5 under the help of twitter web intents.



Introduction to twitter web intents

Web Intents provide popup-optimized flows for working with Tweets & Twitter Users: Tweet, Reply, Retweet, Favorite, and Follow. They make it possible for users to interact with Twitter content in the context of your site, without leaving the page or having to authorize a new app just for the interaction. Web intents are mobile friendly, and super easy to implement.



如果你做过游戏接入网站之类的工作，就知道有时候只是为了实现一个极小的功能，就不得不进行OAuth验证是多么的烦人。而使用twitter的web intents 服务，想要实现发推分享游戏成绩之类的功能，实在是简单到不行。想详细了解的朋友可以看下文档去：

<https://dev.twitter.com/docs/intents>

The usage of twitter web intents

A very simple example of how to use twitter web intents:

Open your browser and type in the following URL

```
https://twitter.com/intent/tweet?text=this is a test
```

you will be redirected to a tweet page:



And in our game we can add a button and then popup a window for tweet while the player click it:

```
window.open(url, "share", "height=300, width=300");
```

With twitter web intents we can lots of cool things in our game and here is just a simplest one among them. You can check the doc: <https://dev.twitter.com/docs/intents> to see what we can do with twitter web intents.

12

Welcome heroku - Prepare for Facebook release

Finally we have finished porting our game and all the testing && debugging stuffs so what's next ?

Bring our game to the WORLD 😎

In order to that we need a place tot host our game, you can do that with your own server for sure but here I'm gonna show you how to deploy own game to a famous game hosting site -- heroku.

The screenshot shows the Heroku homepage with the following content:

- cloud application platform**
deploy and scale powerful apps
- Four icons: **Forget Servers**, **Run Anything**, **See Everything**, and **Trust & Manage**.
- A main section titled "Run and scale any type of app." with the subtext: "Run any web or background process with any web framework or worker type. Get direct, granular control over your app's workloads. Scale distributed processes effortlessly with a single command. Easily scale to millions of users." It includes a link to "Scaling Apps on Heroku".
- A code block showing a Procfile and a terminal command:

```
$ cat Procfile
web: bundle exec rails server -p $PORT
worker: bundle exec rake resque:work QUEUE=*
urgentworker: bundle exec rake resque:work QUEUE=urgent
clock: bundle exec clockwork clock.rb

$ heroku scale web=4 worker=2 urgentworker=1 clock=1
Scaling processes... done
```
- Call-to-action buttons: **How it Works**, **It's free to get started and sign up is instant.**, and **Sign Up**.
- Four footer boxes:
 - We're Hiring**: Makers welcome. Help us change the way app development is done.
 - Critical Production Apps**: The peace of mind you expect, plus 24x7 support, dedicated customer advocates and more.
 - Heroku Partners**: Build your business with experienced service partners and support you can trust.
 - Waza 2013**: Watch the session videos from Heroku's Developer Event.

Introduction to heroku

A introduction from WIKI:

“Heroku is a cloud platform as a service (PaaS) supporting several programming languages. Heroku is owned by Salesforce.com. Heroku, one of the first cloud platforms, has been in development since June 2007, when it supported only the Ruby programming language, but has since added support for Java, Node.js, Scala, Clojure and Python and (un-documented) PHP.”

They have detailed documents easy-to-use command line tools and great technical support .

You can go and check more about heroku from their site:

<http://about.heroku.com/>

Installing heroku

It's easy to install heroku on your PC:

- First go and create a account of heroku
- Then go and download the tools we need to deploy out game -- Heroku Toolbelt
- After installation we can open up the terminal and try “heroku login” to see if heroku tool has been successfully installed
- Enter your account mail and password and we’re done

For more details you can go and check their docs here:

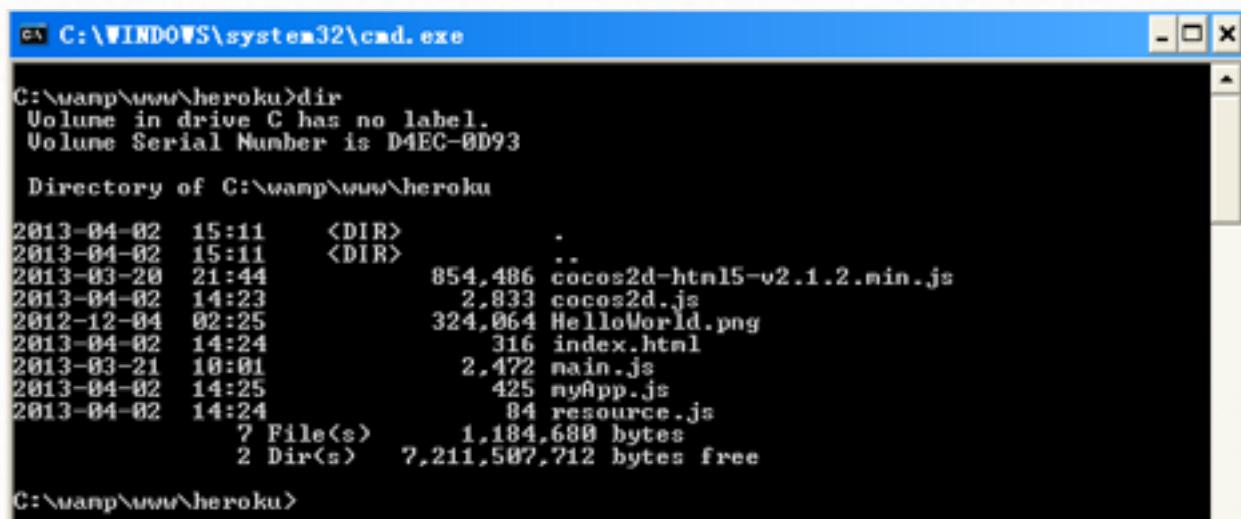
<https://devcenter.heroku.com/articles/quickstart>

Using heroku with git command

We will use git command to work with heroku so if you are not familiar with git you'd better go and learn some basic command at first.

Now we are gonna deploy a very simple cocos2d-html5 project to heroku.

- First create a folder and copy all the files inside:



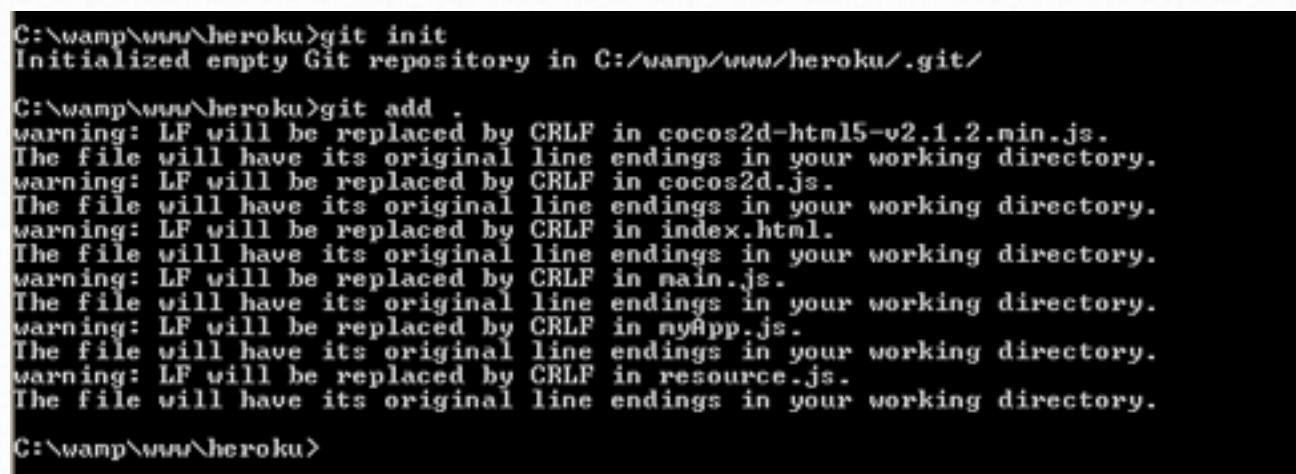
```
cmd C:\WINDOWS\system32\cmd.exe
C:\wamp\www\heroku>dir
Volume in drive C has no label.
Volume Serial Number is D4EC-8D93

Directory of C:\wamp\www\heroku

2013-04-02  15:11    <DIR>      .
2013-04-02  15:11    <DIR>      ..
2013-03-20  21:44        854,486  cocos2d-html5-v2.1.2.min.js
2013-04-02  14:23        2,833  cocos2d.js
2012-12-04  02:25        324,064 HelloWorld.png
2013-04-02  14:24            316  index.html
2013-03-21  10:01        2,472  main.js
2013-04-02  14:25        425  myApp.js
2013-04-02  14:24            84  resource.js
2013-04-02  ? File(s)   1,184,688 bytes
2 Dir(s)   7,211,507,712 bytes free

C:\wamp\www\heroku>
```

- Then initialize a local git repository, add and commit the files:



```
C:\wamp\www\heroku>git init
Initialized empty Git repository in C:/wamp/www/heroku/.git/

C:\wamp\www\heroku>git add .
warning: LF will be replaced by CRLF in cocos2d-html5-v2.1.2.min.js.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in cocos2d.js.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in main.js.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in myApp.js.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in resource.js.
The file will have its original line endings in your working directory.

C:\wamp\www\heroku>
```

```
C:\wamp\www\heroku>git commit -m "first commit"
[master (root-commit) 66a692d] first commit
warning: LF will be replaced by CRLF in cocos2d-html5-v2.1.2.min.js.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in cocos2d.js.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in main.js.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in myApp.js.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in resource.js.
The file will have its original line endings in your working directory.
    7 files changed, 1868 insertions(+)
    create mode 100644 HelloWorld.png
    create mode 100644 cocos2d-html5-v2.1.2.min.js
    create mode 100644 cocos2d.js
    create mode 100644 index.html
    create mode 100644 main.js
    create mode 100644 myApp.js
    create mode 100644 resource.js
C:\wamp\www\heroku>_
```

- Then we need to create a empty app on heroku:

```
C:\wamp\www\heroku>heroku login
Enter your Heroku credentials.
Email: supersuraccoon@gmail.com
Password (typing will be hidden):
Authentication successful.

C:\wamp\www\heroku>heroku apps:create cocoshelloworld
Creating cocoshelloworld... done, stack is cedar
http://cocoshelloworld.herokuapp.com/ | git@heroku.com:cocoshelloworld.git
Git remote heroku added
C:\wamp\www\heroku>_
```

- Push the all files to heroku:

```
C:\wamp\www\heroku>git push heroku master
Counting objects: 9, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 499.32 KiB / 31 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0)

!     Heroku push rejected, no Cedar-supported app detected

To git@heroku.com:cocoshelloworld.git
 ! [remote rejected] master -> master (pre-receive hook declined)
error: failed to push some refs to 'git@heroku.com:cocoshelloworld.git'
C:\wamp\www\heroku>_
```

And that's it now we have just created a app on heroku and push own cocos2d-html5 project on it.

You might run into the following error while trying to push the project:

git push heroku master permission denied (publickey)

To get rid of this error you can first check the ssh-key currently added in heroku:

```
heroku keys
```

if there is no output then we need to add a ssh-key to heroku:

```
heroku keys:add xxx/.ssh/id_rsa.pub
```

If you don't have ssh-key yet then you can create one using the following command:

```
ssh-keygen -t rsa
```

more detailed doc here:

<https://help.github.com/articles/generating-ssh-keys>

If you're under Windows OS then you might need Copssh -- an OpenSSH server and client implementation for Windows systems --to help you to create ssh-keys.

If you still keep getting the error above then you might try the following command:

```
git remote rm heroku  
git remote add heroku git@heroku.com:myGame.git
```

Now we should able to push our game to heroku 😊

Deploying the game as php app

Now since we have learnt how to use git command to work with heroku it's time to run own game from heroku.

We are gonna deploy own game as a php application -- don't worry, that's easy -- we just need a file named index.php and let's rename own index.html to home.html to avoid mistaking the two files (index.html and index.php).

```
C:\wamp\www\heroku>dir
Volume in drive C has no label.
Volume Serial Number is D4EC-8D93

Directory of C:\wamp\www\heroku

2013-04-02  15:18    <DIR>      .
2013-04-02  15:18    <DIR>      ..
2013-03-28  21:44            854,486 cocos2d-html5-v2.1.2.min.js
2013-04-02  14:23            2,833 cocos2d.js
2012-12-04  02:25            324,864 HelloWorld.png
2013-04-02  14:24            316 home.html
2013-04-02  15:19            62 index.php
2013-03-21  10:01            2,472 main.js
2013-04-02  14:25            425 myApp.js
2013-04-02  14:24            84 resource.js
                           8 File(s)   1,184,742 bytes
                           2 Dir(s)   7,208,235,000 bytes free

C:\wamp\www\heroku>
```

Then we need to add one line into the index.php:

```
include_once("home.html");
```

Now when we push own app to heroku it will detect that it's a php application and start necessary service we need to run own application:

```
C:\wamp\www\heroku>git push heroku master
Counting objects: 12, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (12/12), done.
Writing objects: 100% (12/12), 499.58 KiB / 149 KiB/s, done.
Total 12 (delta 1), reused 0 (delta 0)

----> PHP app detected
----> Bundling Apache version 2.2.22
----> Bundling PHP version 5.3.10
----> Discovering process types
      Procfile declares types => <none>
      Default types for PHP    -> web

----> Compiled slug size: 10.0MB
----> Launching... done, v4
      http://cocoshelloworld.herokuapp.com deployed to Heroku

To git@heroku.com:cocoshelloworld.git
 * [new branch]      master -> master

C:\wamp\www\heroku>_
```

Now we can try to run own game from heroku:



```
C:\WINDOWS\system32\cmd.exe
C:\wamp\www\heroku>heroku open
```



That's it 😊

We successfully deployed our game to heroku as php application and we will add authorize function to the index.php later when we are trying to connect our game to Facebook.

13

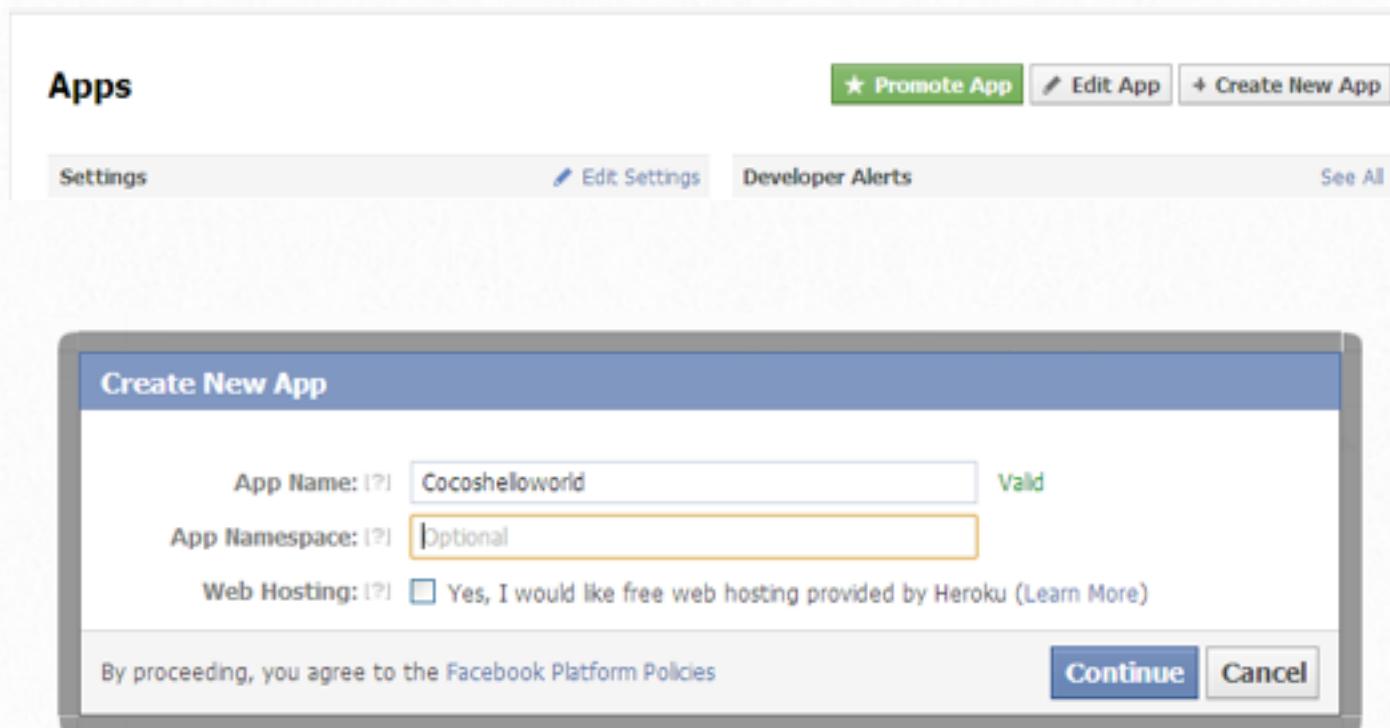
Welcome Facebook - Connect our game to Facebook

Now we have our game deployed to heroku and in this chapter we are gonna connect our game to Facebook so that all people around the world can play our game in Facebook.



Prepare for connecting with Facebook

First we need a Facebook account and then we can create a web game in Facebook developer page:



After creating the game we need fill in some information about our game to help Facebook. To find our game.

Here are some important info you need make sure to right:

App Domains: cocoshelloworld.herokuapp.com -- Enable auth on domain and subdomains

Canvas Page: <http://apps.facebook.com/cocoshelloworld> -- URL of your game on Facebook

Canvas URL: <http://cocoshelloworld.herokuapp.com> -- Content pulled from this base URL to iFrame on Facebook

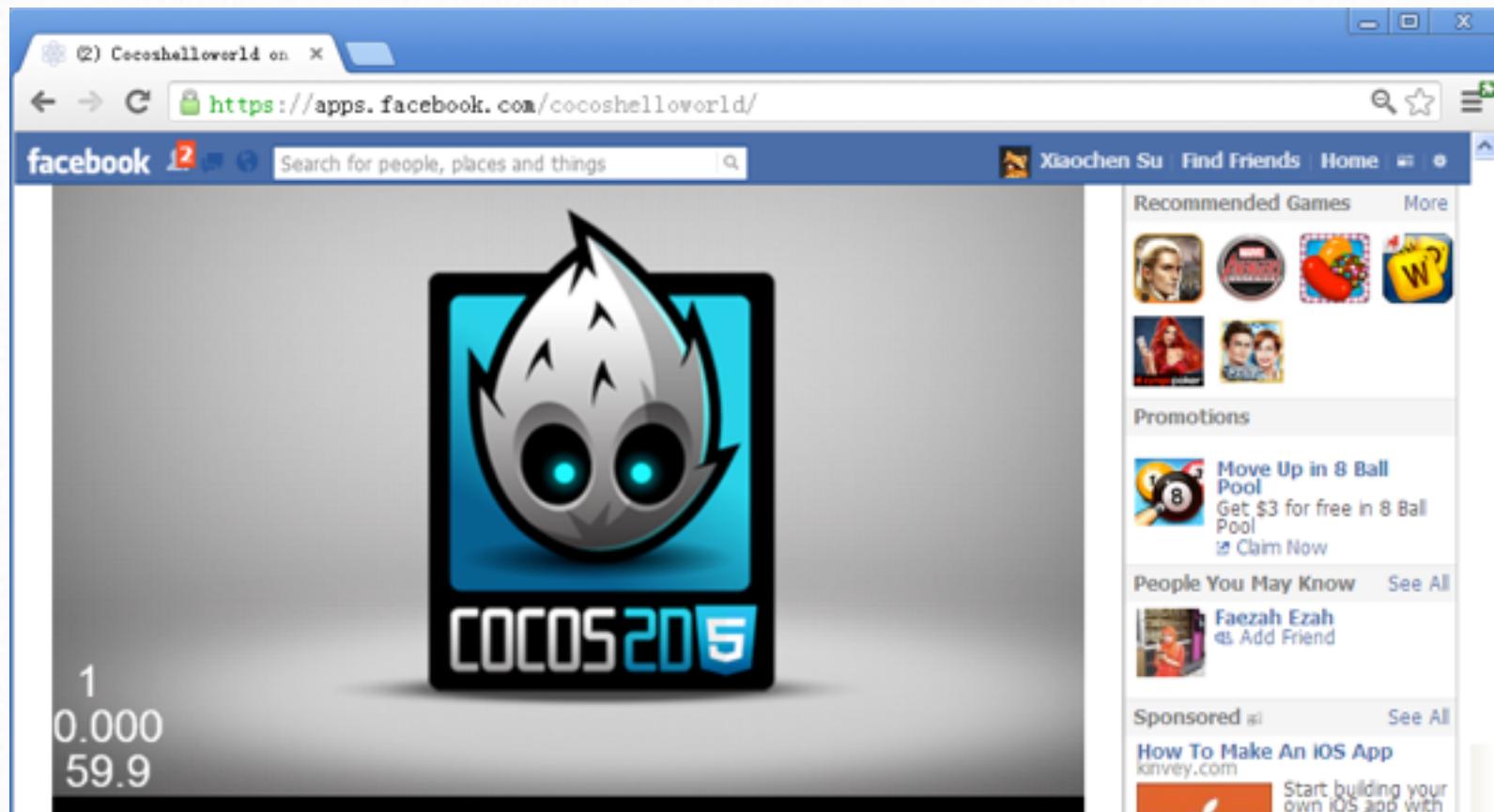
Basic Info

Display Name: [?]	cocoshelloworld
Namespace: [?]	cocoshelloworld
Contact Email: [?]	supersuraccoon@gmail.com
App Domains: [?]	cocoshelloworld.herokuapp.com
Hosting URL: [?]	You have not generated a URL through one of our partners (Get one)
Sandbox Mode: [?]	<input type="radio"/> Enabled <input checked="" type="radio"/> Disabled

Select how your app integrates with Facebook

<input checked="" type="checkbox"/> Website with Facebook Login	X
Site URL: [?]	<input type="text" value="http://cocoshelloworld.herokuapp.com/"/>
<input checked="" type="checkbox"/> App on Facebook	X
Canvas Page: [?]	<input type="text" value="http://apps.facebook.com/cocoshelloworld"/>
Canvas URL: [?]	<input type="text" value="http://cocoshelloworld.herokuapp.com/"/>
Secure Canvas URL: [?]	<input type="text" value="https://cocoshelloworld.herokuapp.com/"/>
Canvas Width: [?]	<input type="radio"/> Fluid <input checked="" type="radio"/> Fixed (760px)
Canvas Height: [?]	<input type="radio"/> Fluid <input checked="" type="radio"/> Fixed at <input type="text" value="600"/> px

Now save the info about our game and you can see our game running by access the Canvas page above.



Authorization with Facebook

Now Facebook knows where our game is and can load it successfully. But that's not done yet there is one more thing that we need to do and that is asking for authorization since we might need to access user's information when they play our game.

The authorization logic is quite straightforward:

- First we check if the has already done the authorization process
- If so we only need to load our game and let him play
- If not then we need show him the authorization dialog asking for his promise
- And if the user refuse the authorization then we won't load the game

First we need download the fb-php-sdk and copy it into our game folder and we will use php for the authorization part.

<https://developers.facebook.com/docs/reference/php/>

```
<body>
<script language="php">
require 'fb-php-sdk/facebook.php';
$app_id = 'xxxxxxxx';
$app_secret = 'xxxxxxxx';
$app_namespace = 'mygame';
$app_url = 'https://apps.facebook.com/' . $app_namespace . '/';
$scope = 'email,publish_actions';
$facebook = new Facebook(array(
    'appId' => $app_id,
    'secret' => $app_secret,
));

```

And here is how we do it in index.php file:

```
// Get the current user
$user = $facebook->getUser();
// If the user has not installed the app, redirect them to the Login Dialog
if (!$user) {
    $loginUrl = $facebook->getLoginUrl(array(
        'scope' => $scope,
        'redirect_uri' => $app_url,
    ));
    print('<script> top.location.href=' . $loginUrl . "'</script>');
}
else {
    include_once("home.html");
}
</script>
</body>
```

The code is not difficult to understand. Now update our game on heroku and his time when we try to access our game on Facebook you'll first see a dialog pops up asking for authorization:



When we agree with the authorization we can see it in our app region:

Your Apps

Sort by: **Last Used** ▾

 cocoshelloworld Used less than 24 hours ago Use Now · Settings	 LinksAlpha.com Used September 23 Visit Website · Settings
 Dropbox Used June 10 Visit Website · Settings	 Bing Used more than 6 months ago Visit Website · Settings
 BlogGlue Used more than 6 months ago Visit Website · Settings	 Diamond Dash Used more than 6 months ago Play Now · Settings

Playing with Dialogs in Facebook

Now we have passed the authorization part let's try to use the Facebook API to share the score info within our game.

<https://developers.facebook.com/docs/reference/dialogs/>

- First we need import the js file we need

```
<script src="//connect.facebook.net/en_US/all.js"></script>
```

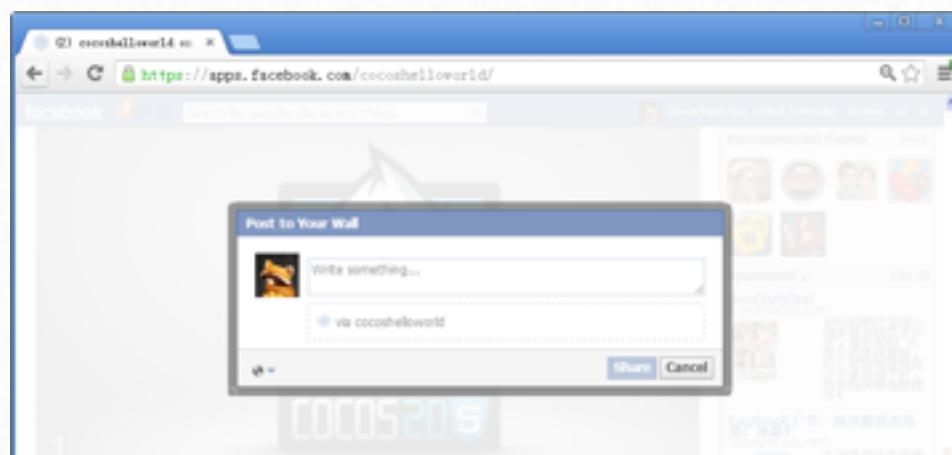
- Then we can need initialize it:

```
FB.init({appId: "YOUR_APP_ID", status: true, cookie: true});
```

- Then we can use the Facebook API at ease:

```
FB.ui({ method: 'feed',
  caption: caption,
  picture: 'xxx.png',
  name: 'XXXXXXXXX'
}, null);
```

And now you can see the Feed Dialog pops up:



Decorate our game page

Now we finished connect our game to Facebook but you can see there is only one canvas in our game page and that dose not look good.

So let's add a background image to our game page.

That's easy with a simple HTML code:

```

```

But I had problem when tried to put this in home.html, the canvas shows after the image instead of right on it.

I solved this by putting the image into the index.php file or put the canvas and image in a div and write some CSS t set the z-index of both:

```
<div id="container">
  
  <canvas id="gameCanvas" width="320" height="480"></canvas>
</div>

body{text-align: center;}.img{position:absolute;z-index:1;}

#container{
  position:relative;
}
#gameCanvas{
  position:relative;
  z-index:20;
}
```

Now it looks much better but I am sure you can do far more better than I do.

Finally we connect our game to Facebook and then implemented the authorization part. Now we can submit our game for reviewing.

We only asked for authorization in the simple demo above actually after authorization Facebook will pass the user info to you and there are lots of things you can do with that, you can explore this by yourself:

<https://developers.facebook.com/docs/tutorials/canvas-games>

Now we have Facebook covered let's go and take over Sina Weibo in the next two chapter



14

Welcome SinaAppEngine - Prepare for Sina Weibo release

Now we got Facebook covered it's time for Sina Weibo.

We have already had the experience of deploying our game to heroku and connect it to Facebook so you mush feel quite familiar with what we gonna do in the next two chapter --deploy our game on sina app engine and connect it to Sina Weibo.

Introduction to Sina App Engine

SAE is a leading PaaS (Platform-as-a-Service) platform in China, it not only supports PHP development language but also takes the lead in supporting Java development language.

The screenshot shows the SinaAppEngine website. At the top right, there are links for "简体中文" (Simplified Chinese), "Activity", "Register", and "Login". Below the header is a navigation bar with links for "Home", "Architecture", "Showcase", "Pricing", "Docs", "Services", "Community", and "Guide". A search bar with the placeholder "输入想要搜索的关键字" (Input the keyword you want to search) and a "搜" (Search) button are also present. The main content area features a title "Development and running platform of distributed web service" and a grid of eight service icons: Load Balancer (with a bar chart icon), Web Servers (with a server icon), Cache (with a speedometer icon), Database (with a database icon), Security (with a shield icon), Task Schedulers (with a clock icon), Storage (with a cloud icon), and Extension (with a puzzle piece icon). Each service has its English name and Chinese name listed below the icon.

简体中文 | Activity Register Login

SinaAppEngine

输入想要搜索的关键字 搜

Home Architecture Showcase Pricing Docs Services Community Guide ▶

Development and running platform of distributed web service

负载均衡 Load Balancer	Web服务器 Web Servers	缓存 Cache	数据库 Database	安全 Security	任务调度 Task Schedulers	存储服务 Storage	扩展服务 Extension
-----------------------	-----------------------	-------------	-----------------	----------------	-------------------------	-----------------	-------------------

Sina App Engine code management

Sina app engine uses svn for code deployment. We are gonna use torero svn to deploy our code.

- First we need to create an account
 - Then we need to create an app and choose php as language

创建应用

创建空应用 选择开发框架 不懂技术？一键安装云商店热门应用

AppID

* 二级域名 (AppID) .sinaapp.com (允许由数字,字母组成,长度为4到18位)
唯一标识,也是二级域名前缀, 创建后不可修改。

App Name

* 应用名称
应用的中文名称,供显示用。

* 验证码

App Info

应用描述

App Language

开发语言 PHP Python Java 应用创建后,开发语言不可修改

App Type

应用类型 Web应用 移动应用

[创建应用](#)

应用列表

苏麒麟
未认证
申请开发者认证

我创建的应用 [如何部署代码？](#)

[安装应用](#) [创建新应用 \(2/10\)](#)

⚠ 因未对SVN、二级域名等服务收费，造成资源滥用；为避免资源滥用，给您提供更优质的服务，从2012年10月30日起，新浪云开始收取应用租金。您可以通过以下方式减免租金： [已通过实名认证](#) [申请开发者认证](#) [已通过手机复验](#)

[php](#) ClearFiveStages <http://clearfivestages.sinaapp.com>

[php](#) cocos2dhtml5elloworld

- Then we need to create a version for our game in the code manager tab

html5helloworld > 代码管理

应用信息

- 汇总信息
- 预算设置
- ... 2013-4-2 14:29:20

设为默认 版本 链接 xhprof 调试 错误显示 操作

1部署于 2013-04-02 14:29:20 http://1.html5helloworld.sinaapp.com 点击开启1小时 点击关闭 编辑代码 | 上传代码包 | 剔除

Create Version

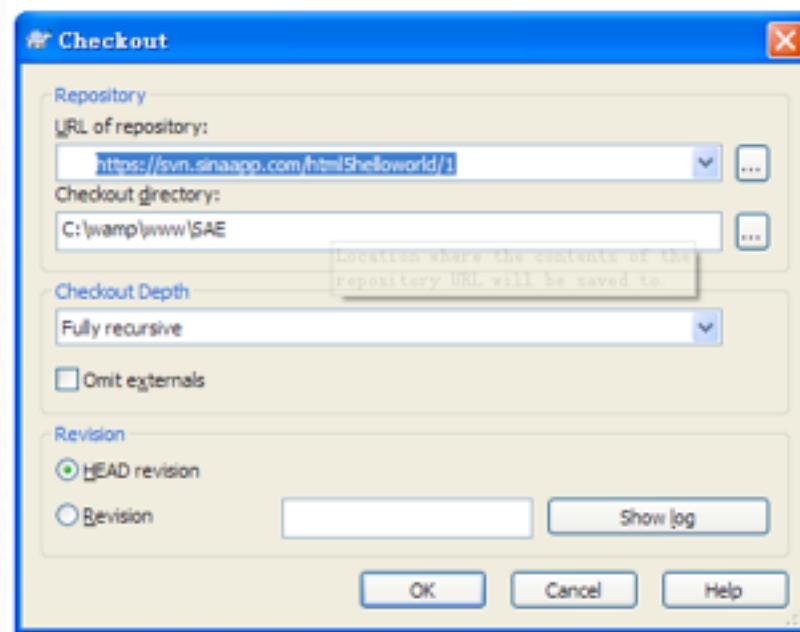
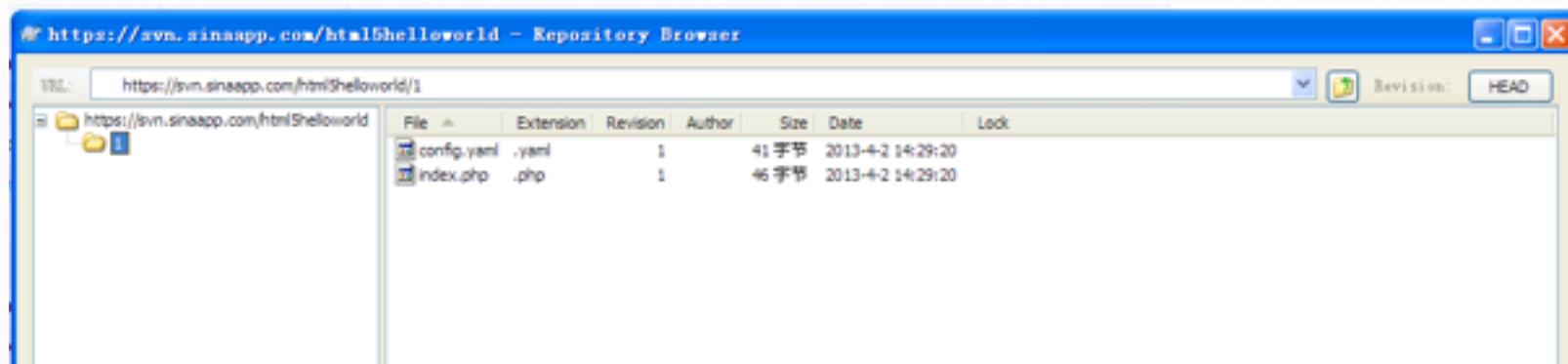
创建版本

请输入版本号 1

Version Number

大小不超过50M。
只允许存在10个以内的版本，并且版本号必须为正整数。（也就是说appname下面只允

- Now we can use our svn tool to checkout the code

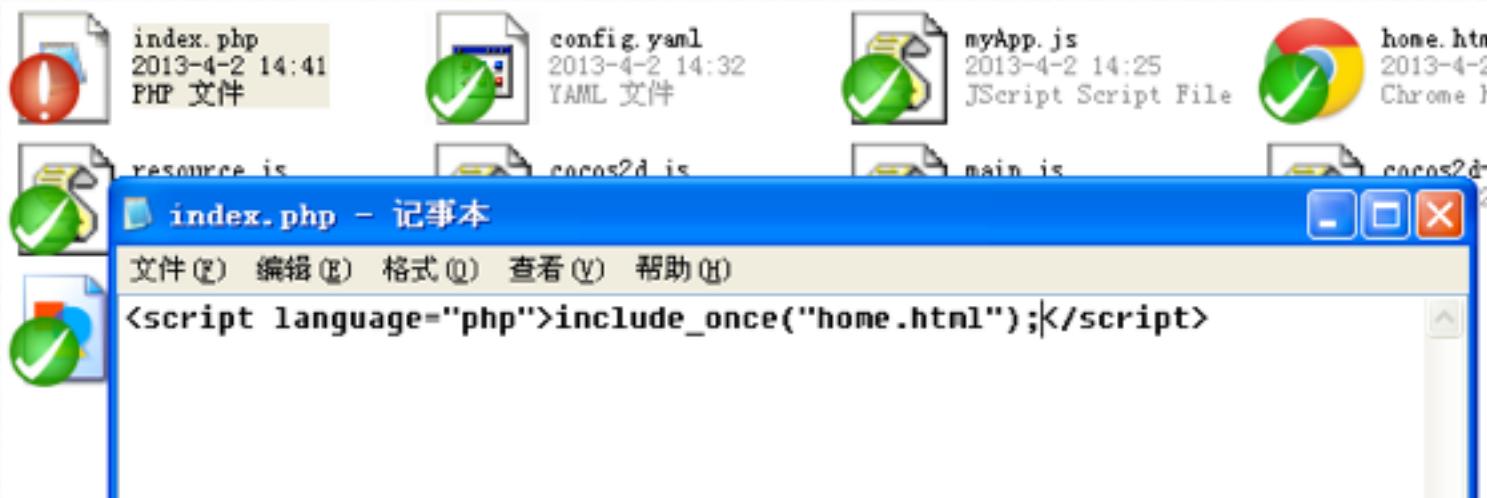


Now we have checked out the project and we'll distribute our game to Sina App Engine in next section 😞

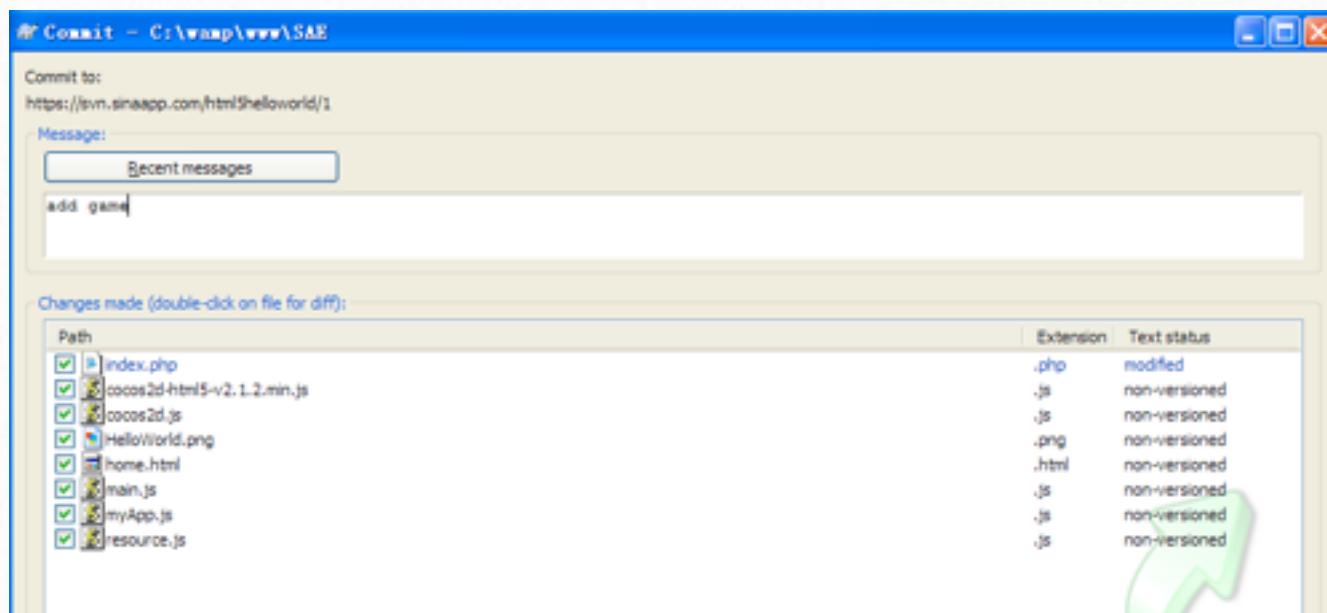
Deploying the game as php app

Now what we are gonna do the same thing as we did when deploy to heroku.

We need to rename our index.html to home.html and Create a new file named index.php



Then we submit our changes to Sina App Engine.



And that's it:



15

Welcome Sina Weibo - Connect our game to Sina Weibo

Connect our game to Sina Weibo is almost the same as we did with Facebook. We need to tell Sina Weibo where our game is and handle the authorization with Sina Weibo.



Prepare for connecting with Sina Weibo

- First we need an account and create an app and fill in the required info:

首页 > 应用开发 > 创建站内应用

创建站内应用

App Name 应用名称: ✓

App Info 应用简介: ✓

App Detail 应用介绍: ✓

Domain Bind 域名绑定: 是 否 应用安全选项，绑定

Tag 标签: ✓

我已阅读并接受《新浪微博开发者协议》

[申请SAE应用托管服务](#)

[创建](#) [取消](#)

- Then we can need to tell Sina Weibo where to find our game:

Canvas Page
站内应用地址: ✓

Canvas URL
应用实际地址: ✓

iFrame Height
iframe高度: px ✓

高度自适应

Canvas Page: Our game URL in Sina Weibo

Canvas URL: Where our game is -- Sina Weibo will pull content from this URL

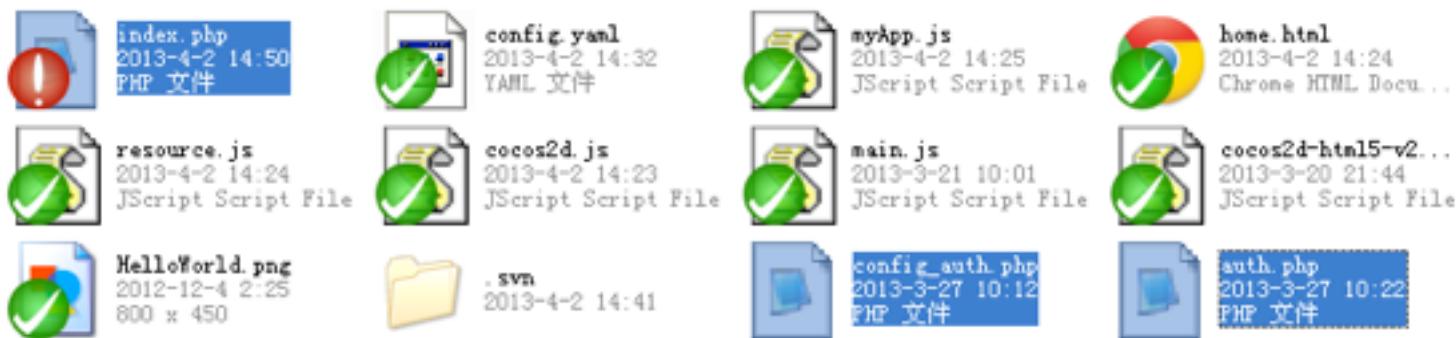
Now go and check our game with the Canvas Page URL



Authorization with Sina Weibo

Authorization with Sina Weibo is similar with we did in Facebook.

- First we need to modify the index.php



```
<body>
<script language="php">
include_once 'config_auth.php';
include_once 'saetv2.ex.class.php';
if(!empty($_REQUEST["signed_request"])){
    $o = new SaeTOAuthV2( WB_AKEY , WB_SKEY );
    $data=$o->parseSignedRequest($_REQUEST["signed_request"]);
    if($data=='-2') {die('error');}
    else {$_SESSION['oauth2']=$data;}
}
if (empty($_SESSION['oauth2'][ "user_id" ])) {
    include "auth.php";
    exit;
}
else {
    include_once("home.html");
}
</script>
</body>
```

And add a file named auth.php:

```
<script src="http://tjs.sjs.sinajs.cn/t35/apps/open/js/frames/client.js"
language="JavaScript"></script>
<script>
function authLoad(){
App.AuthDialog.show({
client_id : 'xxxxxx',
redirect_uri : 'http://apps.weibo.com/cocoselloworld',
height: 120
});
}
</script>
<body onload="authLoad();">
</body>
```

and config_auth.php with config info:

```
<?php
define( "WB_AKEY" , 'xxxxxx' );
define( "WB_SKEY" , 'xxxxxx' );
define( "CANVAS_PAGE" , "http://apps.weibo.com/cocoselloworld" );
?>
```

Now when we load our game the authorization dialog will pop up:



Playing with Weibo publisher

Weibo publish provides us features such as sending message with picture, emoji and topic from our app by adding only a few code to our app.



To enable Weibo publisher in our game we need to:

- Import xmlns in home.html:

```
<html xmlns:wb="http://open.weibo.com/wb">
```

without this the publisher might pops up but you will get the “error sending message”.

- Import wb.js in home.html:

```
<script src="http://tjs.sjs.sinajs.cn/open/api/js/wb.js?appkey=xxxxxx" type="text/javascript" charset="utf-8"></script>
```

- Make sure we have set the “bind domain” in

OAuth2.0 授权设置

编辑

授权回调页： <http://apps.weibo.com/cocoselloworld>

取消授权回调页： 未填写

安全设置

Need Bind Domian?

域名绑定： 是 否

YES **NO**

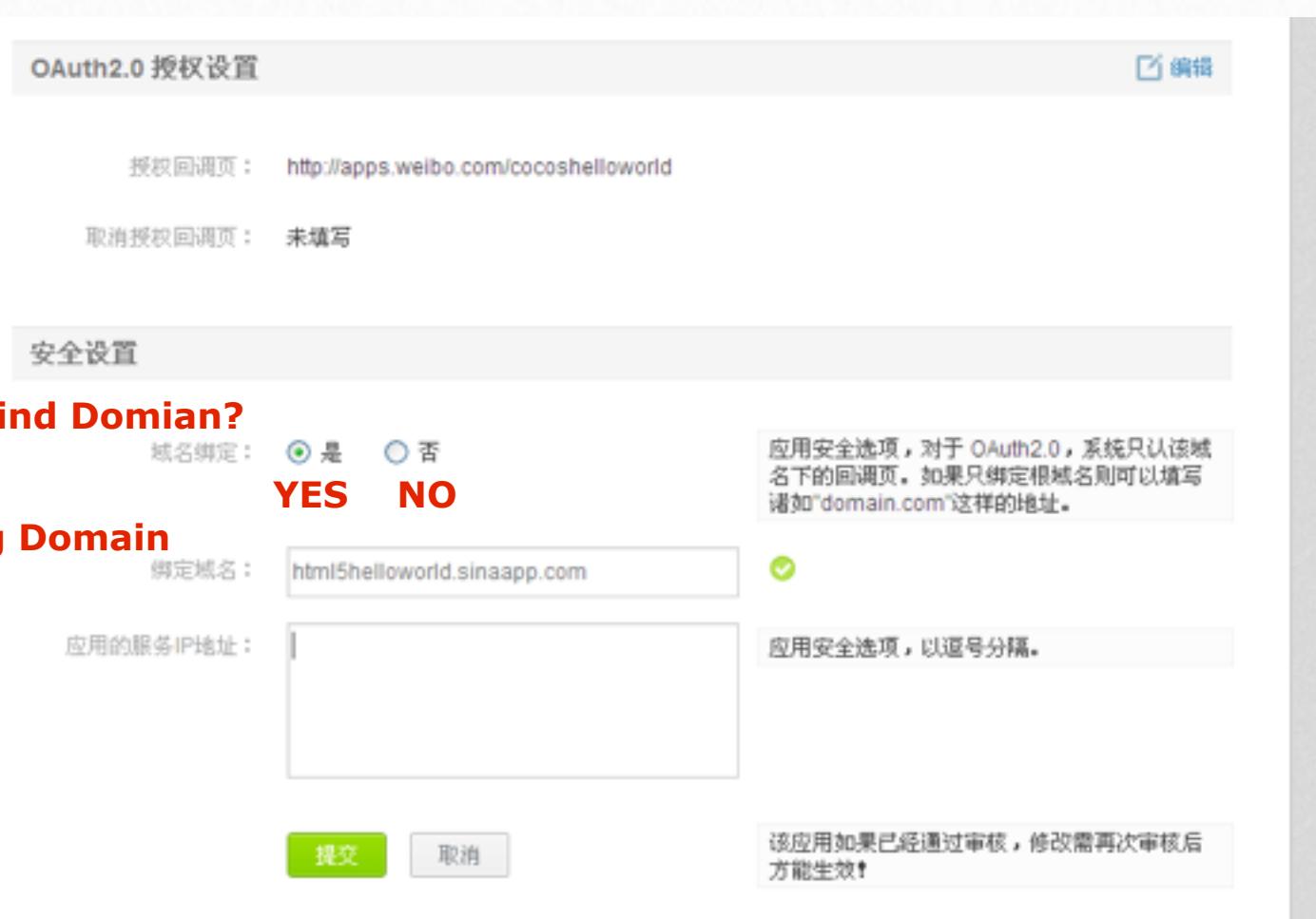
Binding Domain

绑定域名：

应用的服务IP地址： 应用安全选项，以逗号分隔。

提交 **取消**

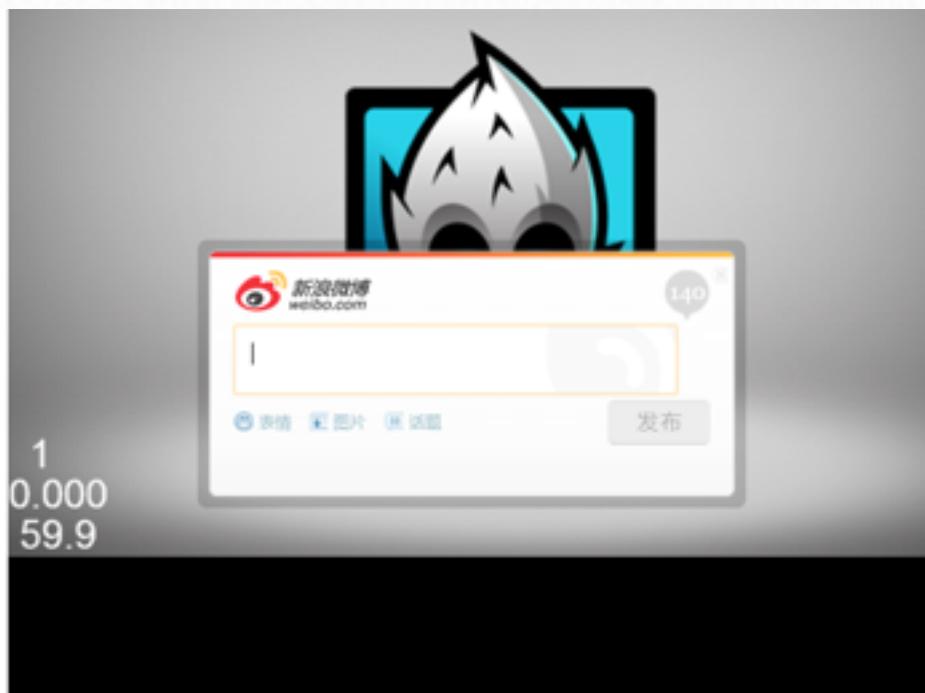
该应用如果已经通过审核，修改需再次审核后方能生效！

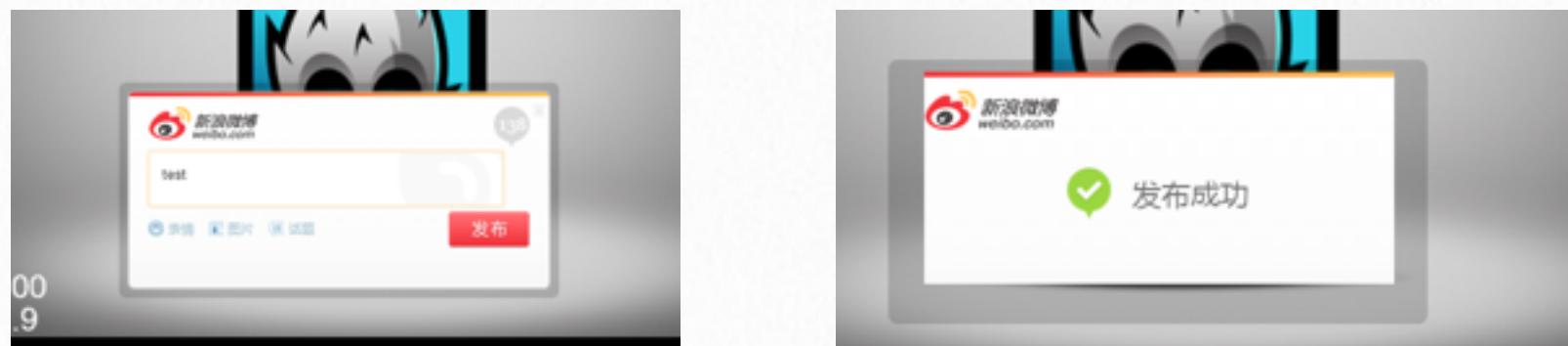


- Now we can use Weibo publisher inside our game like this:

```
WB2.anyWhere(function(W){W.widget.publish({id : ''});});
```

And finally we can use Weibo publisher to send message:





Weibo publisher supports a lot of parameters you can play with, check out the docs:

<http://weibojs.com/widget/publish.php#wb>

Now we have successfully connected our game to Sina Weibo :)

Sina Weibo also provides lots of API you can use to implement something cool in our game.

16

My site my game - Embedding the game into our site

One of the cool thing with our html5 game is that we can depoly it to a lot of places -- say our own site.

You can embed your game in your article so that your visitors can read while enjoying your game.

What I'll show you in this chapter is how to embed the simple game we deployed to Facebook and Sina Weibo in our post in a Wordpress blog -- you should notice that the way embedding game inside a post might vary according to the blogging tool you're using.

SuperSuRaccoon's World Cocos2d game develop (Know-how, Demo, Tutorial)

The screenshot shows a blog post titled "[Sticky] ios-cocos2d develop Recommended Blog List (Keep Updating)". It includes a "Continue reading →" link, a "Choose Language" dropdown set to English, and a "my LinkedIn profile" button. To the right, there are links for "FingerEscaper On AppStore" and "Clear-5-Stage On Appstore", each with a small thumbnail image. At the bottom, there's a placeholder for "Your AD Here :)".

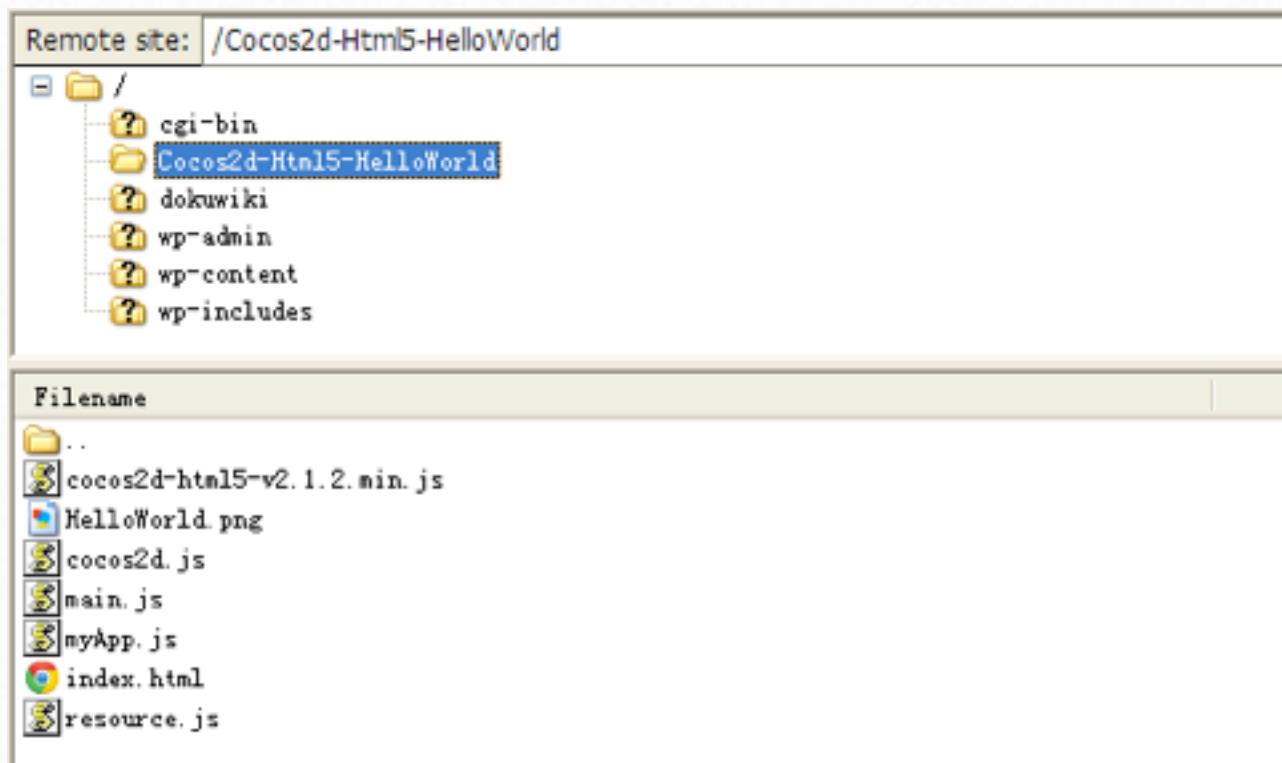
[Sticky] ios-cocos2d develop Recommended Blog List (Keep Updating)
Posted on Saturday February 11th, 2012 by SuperSuRaccoon
[Continue reading →](#)
Posted in cocos2d, Develop, Resource | 16 Comments | Edit

[Sticky] Willing to share your own demo with the world? :)
Posted on Monday October 3rd, 2011 by SuperSuRaccoon
[Continue reading →](#)
Posted in Announcement | Leave a comment | Edit

[Sticky] Resource Download List... (Keep Updating)

Embedding game in a post

- First we need to upload our game to the site using FTP:



- Then we create a post to embed our game in:

Title (English)
test

Permalink: <http://www.supersuraccoon-cocos2d.com/2013/04/02/testztestjatest/> [Edit](#) [Get Shortlink](#)

Title (中文)
test

Title (日本語)
test

Upload/Insert English 中文 日本語 Visual HTML

`<div style="display:inline-block; width:auto; margin: 0 auto; background: black; position:relative; border:5px solid black; border-radius: 10px; box-shadow: 0 5px 50px #333">
 <canvas id="gameCanvas" width="480" height="320"></canvas>
</div>
<script src="/ClearFiveStages-Deploy/parse-1.2.0.min.js"></script>
<script src="/ClearFiveStages-Deploy/boot-html5.js"></script>`

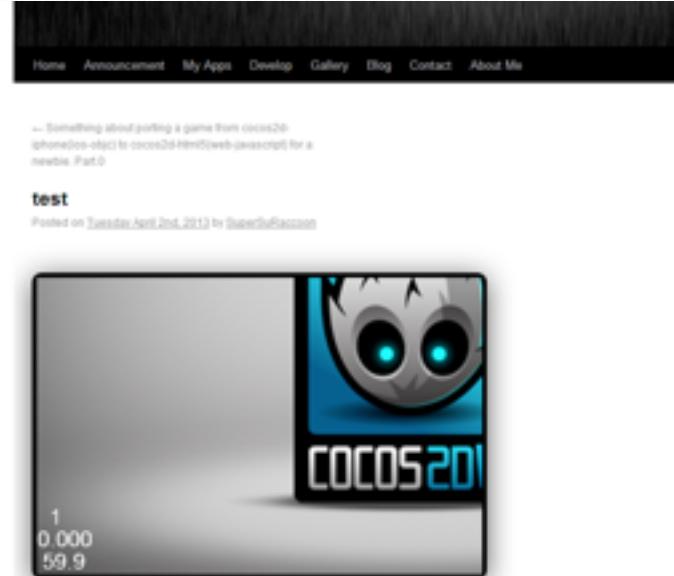
Then save and preview the post and you can see the error message of bad resource path:

```
⚠ Resource interpreted as Image but transferred with MIME type text/html: "http://googleads.g.doubleclick.net/pagead/adview?ai=CP5Pe-oJauZCzHce9igf_vL_1auHBrKoiH2M3ua-NgiAxT56PjV1lk_Bua01Y8Z2WABgbImTygvHz_aQ&sig=Pm-HTQzrV7g".  
↳ XHR finished loading: "https://plusone.google.com/_scs/apps-static/_/js/k=oz_gapi.en.a4fuhuQAGTY..  
sv=1/d=1/am=00/rq=AfTR5TP7eJ09ja85G1_T0N1Iv9F00057g/cb=gapi.loaded_0".  
↳ XHR finished loading: "https://plusone.google.com/_scs/apps-static/_/js/k=oz_plusone.en_US.qVwvnR..  
sv=1/d=1/am=CA/rq=AfTR5TP9KxV9UfV5IXuEc45ndelvdixNYYA/cb=gapi.loaded_1".  
↳ XHR finished loading: "http://www.supersuraccoon-cocos2d.com/wp-content/plugins/wp-share-list/is/sharelistData.js".  
① GET http://www.supersuraccoon-cocos2d.com/cocos2d-html5-v2.1.2.min.js 404 (Not Found)
```

So we need to change the resource path:

```
s._super(a),visit:function(){var a=this._absolutePos  
nction(){return new cc.ParallaxNode};(function(){var  
olExtension/CCScale9Sprite.js ..extensions/GUI/CCCon  
,e=a.concat(c.appFiles);e.push("main.js");if(-1<navig  
pabilities.mp3=="no"!=a.canPlayType("audio/mpeg")&&"!  
decs\x3d"1")&&"!=a.canPlayType('audio/wav; codecs\x  
&!this.soundList.hasOwnProperty(c)){var d=new cc.SF  
  
tag:'gameCanvas', //the dom element to run cocos2d on  
//engineDir:'../cocos2d/',  
SingleEngineFile:'/Cocos2d-Html5-HelloWorld/cocos2d-html5-v2.1.2.min.js',  
appFiles:[  
    '/Cocos2d-Html5-HelloWorld/resource.js',  
    '/Cocos2d-Html5-HelloWorld/myApp.js'//add your own files in order here  
]  
  
var s_HelloWorld = "/Cocos2d-Html5-HelloWorld>HelloWorld.png";  
var g_ressources = [  
    {src:s_HelloWorld}  
];
```

Now everything works just fine 😊



17

The magic of HTML5 - Embedding the game into iBook

Frankly speaking, I wasn't planning writing this in the first place or rather I don't even know it is possible to get this done.

I happened to have the thought that it must be pretty cool if we can embed the game into the book as widget.

So I googled some related knowledge and had a little experiment and to my astonishment we can get this done within one minute -- yep that's right -- ONE MINUTE 

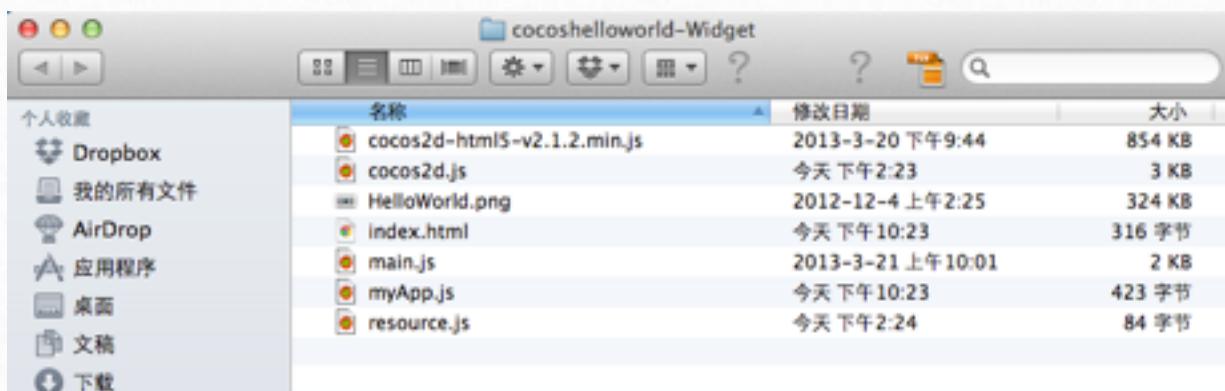
Changing our game to widget

As you know that there're lots of widgets in dashboard under MacOS such as calculator, notepad, calendar etc.

And apple also provides us a tool to develop widget that is -- Dashcode.

But we do not even need to use the Dashcode for our game widget -- a notepad app will be just enough.

- First we create a folder and put all the files in:



- Then we add two files:

Default.png: the image that will appear on the page, and that users will tap to open your widget.

Info.plist: contains all of the information that Dashboard needs to run a widget.

名称	修改日期
cocos2d-html5-v2.1.2.min.js	2013-3-20 下午9:44
cocos2d.js	今天下午2:23
Default.png	2012-12-4 上午2:25
HelloWorld.png	2012-12-4 上午2:25
index.html	今天下午10:23
Info.plist	前天下午7:12
main.js	2013-3-21 上午10:01

- Modify the Info.plist according to our game:

```
<key>CFBundleIdentifier</key>
<string>com.user.widget.cocoselloworld</string>
<key>CFBundleName</key>
<string>CocosHelloWorld</string>
<key>CFBundleDisplayName</key>
<string>CocosHelloWorld</string>
<key>CFBundleShortVersionString</key>
<string>1.0</string>
<key>CFBundleVersion</key>
<string>1.0</string>
<key>CFBundleDevelopmentRegion</key>
<string>English</string>
<key>MainHTML</key>
<string>index.html</string>
<key>CloseBoxInsetX</key>
```

- Rename the folder with extension of “.wdgt”:



- Double click the .wdgt file and install it to Dashboard:



- And you can see our game running in the Dashboard as widget:



Then embedding the game into iBook is piece of cake -- just drag the widget in and that it:



Incredible isn't it ? This really shows the unlimited possibilities of cocos2d-html5, you can just grab and put it in a book in a few seconds to have your readers enjoy the game in a different way 😊

18

The destination - Looking back the trip

Finally we reached our destination 😊

We ported our game from cocos2d-iphone to cocos2d-html5, then we tested and debug it with different kinds of ways, packed it, deployed it to heroku and Sina App Engine, then connected it to Facebook and Sina Weibo, and we even embedded it to our website post and our book 😊

Frankly speaking I did not think I am able to get all these done since this need all different kinds of knowledge and that's not all belong to the programming language category. And what's more I have no experience with javascript, html, php or css before this trip.

But on the other hand, I'm quite confident that I can get this done since I know I am good at and enjoy setting up goals, searching, making guess, deducing, getting close to the goal and searching and making guess, ... till I reach the destination.

I had lots of problems that drove me crazy along the way but solving these did helped me improve my self -- not only enlarged my knowledge but also enhanced my searching ability so I must say it's been a worthwhile trip 😊

There are no high level stuffs like I mentioned in the beginning of the book but I do think it will help some guys wanna take the same trip or are already on their way.

And I want to thank my grandmother here since she taught me about this game when I was a liitle child and it's her 80 years old birthday this May and I hope this will be a good present for her ae well.

The game I mentioned in this book and the demo project will all be open sourced and you can download them from my site: <http://www.supersuraccoon-cocos2d.com/>

Last but not least I wanna thank those who read this book from top to bottom and I hope this book helped you in a certain degree. And I hope more and more developers will join the cocos2d-html5 family since I am a strong believer in cocos2d-html5 -- it's gonna be better and better (at the time writing this book the latest cocos2d-html5 adds webGL support ).

I wish all your games will be heading to the world with help of cocos2d-html5.