

课程安排，请关注微信公众平台或者官方微博

编程语言： **Golang** 与 **html5**

编程工具： **Goland** 和 **HBuilder**

预计平均一周左右更新一或二节课程

授人以鱼，不如授人以渔。

大家好，

欢迎来到 字节教育 课程的学习

字节教育官网：[www.ByteEdu.Com](http://www.ByteEdu.Com)

腾讯课堂地址：[Gopher.ke.qq.Com](http://Gopher.ke.qq.Com)

技术交流群： 221 273 219

微信公众号： **Golang** 语言社区

微信服务号： **Golang** 技术社区

## 目录：

第一季	微信小游戏基础、进阶、提高课程.....	2
第二节	对引擎的支持.....	2
1、	课程内容.....	2
2、	课后作业—难度：★ ☆ ☆ ☆ ☆ .....	6
3、	微信公众平台及服务号 .....	6

# 第一季 微信小游戏基础、进阶、提高课程

## 第三节 对引擎的支持

### 1、课程内容

#### 支持情况



许多开发者对小游戏对 Cocos、Egret、Laya、Unity 等游戏引擎的支持情况非常关心。但是小游戏是一个不同于浏览器的 JavaScript 运行环境,没有 BOM 和 DOM API。然而,基本上所有基于 HTML5 的游戏引擎都是依赖浏览器提供的 BOM 和 DOM API 的。所以如果要在小游戏中使用引擎,需要对引擎进行改造。

目前,Cocos、Egret、Laya 已经完成了自身引擎及其工具对小游戏的适配和支持,对应的官方文档已经对接入小游戏开发做了介绍。

- Cocos : <http://docs.cocos.com/creator/manual/zh/publish/publish-wechatgame.html>
- Egret : <http://developer.egret.com/cn/github/egret-docs/Engine2D/minigame/introduction/index.html>
- LayaBox : <https://ldc.layabox.com/doc/?nav=zh-as-5-0-1>

Unity 目前还没有对小游戏进行适配。不过小游戏提供了对大部分 Canvas 2d 和 WebGL 1.0 特性的支持,支持情况参见 [RenderingContext](#),有能力的开发者可以尝试自行适配。

## 小游戏是一个不同于浏览器的运行环境

无论是怎样的引擎，最终在游戏运行时所做的大部分事情都是 **随着用户的交互更新画面和播放声音**。

小游戏的开发语言是 JavaScript，那么在引擎的底层就需要通过 JavaScript 调用绘制 API 和音频 API。

一段 JavaScript 代码在运行时可以调用的 API 是依赖于 **宿主环境** 的。我们最常用的 `console.log` 甚至都不是 JavaScript 语言核心的一部分，而是浏览器这个宿主环境提供的。常见的宿主环境有浏览器、Node.js 等。浏览器有 BOM 和 DOM API，而 Node.js 则没有；Node.js 有 `fs`、`net` 等 Node.js 核心模块提供的文件、网络 API，而浏览器则不具备这些模块。例如，下面这段在浏览器中可以正常运行的代码，在 Node.js 中运行就会报错。

```
let canvas = document.createElement('canvas')
```

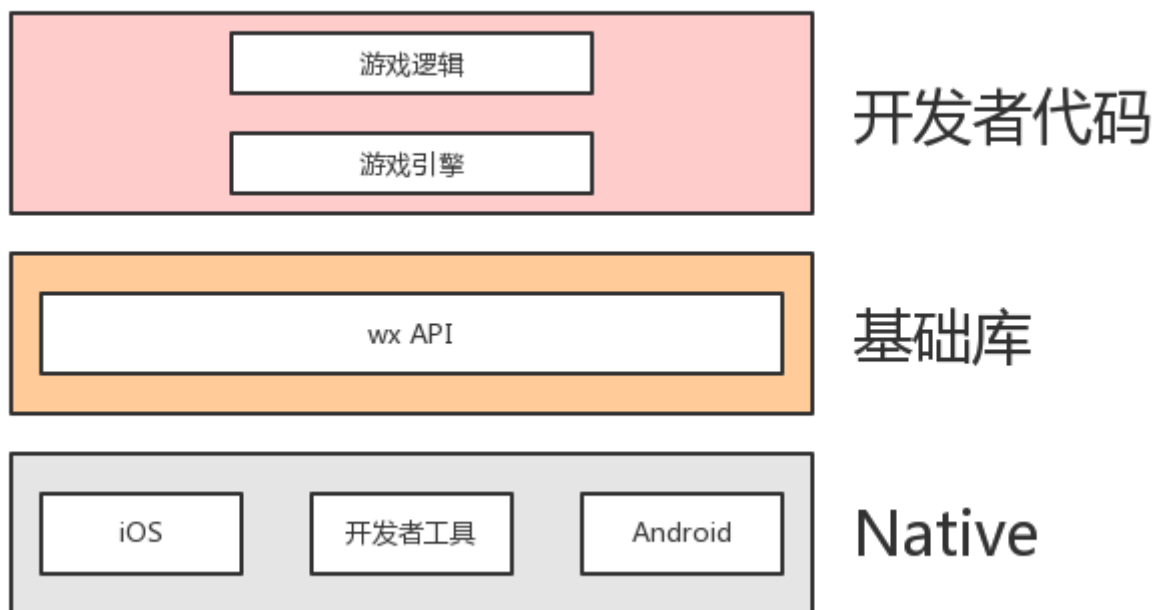
因为 Node.js 这个宿主环境根本没有提供 `document` 这个内置的全局变量。

```
ReferenceError: document is not defined
```

小游戏的运行环境是一个不同于浏览器的宿主环境，没有提供 BOM 和 DOM API，提供的是 **wx API**。

通过 **wx API**，开发者可以调用 Native 提供的绘制、音视频、网络、文件等能力。





如果你想创建画布，你需要调用 [wx.createCanvas\(\)](#)

```
let canvas = wx.createCanvas()
let context = canvas.getContext('2d')
```

如果你想创建一个音频对象，你需要调用 [wx.createInnerAudioContext\(\)](#)

```
let audio = wx.createInnerAudioContext()
// src 地址仅作演示，并不真实存在
audio.src = 'bgm.mp3'
audio.play()
```

如果你想获取屏幕的宽高，你需要调用 [wx.getSystemInfoSync\(\)](#)

```
let { screenWidth, screenHeight } = wx.getSystemInfoSync()
```

但是基于 HTML5 的游戏引擎会通过以下方式去创建画布、音频，获取屏幕宽高

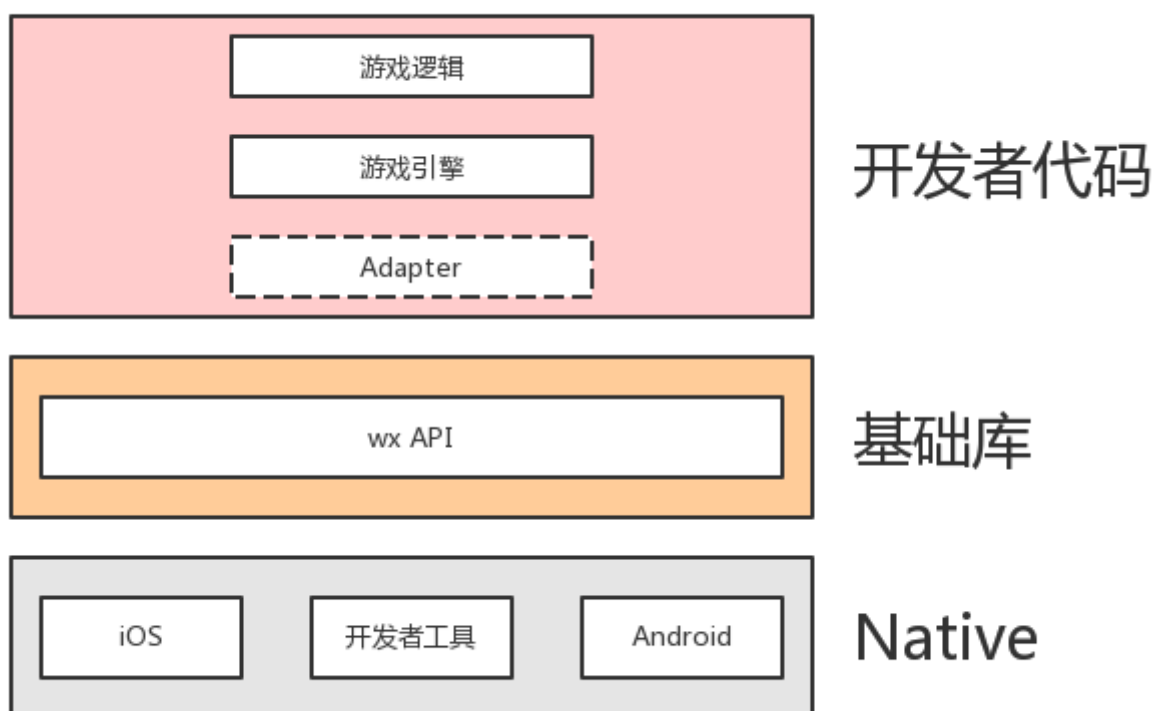
```
let canvas = document.createElement('canvas')
let audio = document.createElement('audio')
console.log(window.innerWidth)
console.log(window.innerHeight)
```

此时会产生错误，理由如前文所述，小游戏这个宿主环境根本没有提供 `document` 和 `window` 这两个在浏览器中内置的全局变量。因为小游戏环境是一个不同于浏览器的宿主环境。

```
ReferenceError: document is not defined
ReferenceError: window is not defined
```

所以，基本上所有基于 HTML5 的游戏引擎都不能直接迁移到小游戏中使用，因为引擎可能或多或少都用到了 BOM 和 DOM 这些浏览器环境特有的 API。只有对引擎进行改造，将对 BOM 和 DOM API 的调用改成 wx API 的调用，引擎才能运行在小游戏环境中。

除了修改引擎，还有一种适配方式，即在引擎和游戏逻辑代码之间加一层模拟 BOM 和 DOM API 的适配层，我们称之为 Adapter。这层适配层在全局通过 wx API 模拟了引擎会访问到的那部分 `window` 和 `document` 对象的属性和方法，使引擎感受不到环境的差异。



Adapter 是用户代码，不是基础库的一部分。关于 Adapter 的介绍，参见教程 [Adapter](#)。

## 2、课后作业—难度：★ ☆ ☆ ☆ ☆

问题交流的平台：

社区问答系统回答：Golang 技术社区 （服务号菜单栏）

---

## 3、微信公众平台及服务号



Golang 语言社区



Golang 技术社区