

学习目标

- 1、通过这个小游戏，进一步熟悉面向对象的代码书写方式
- 2、入门 H5 小游戏，以后上班多一个选择

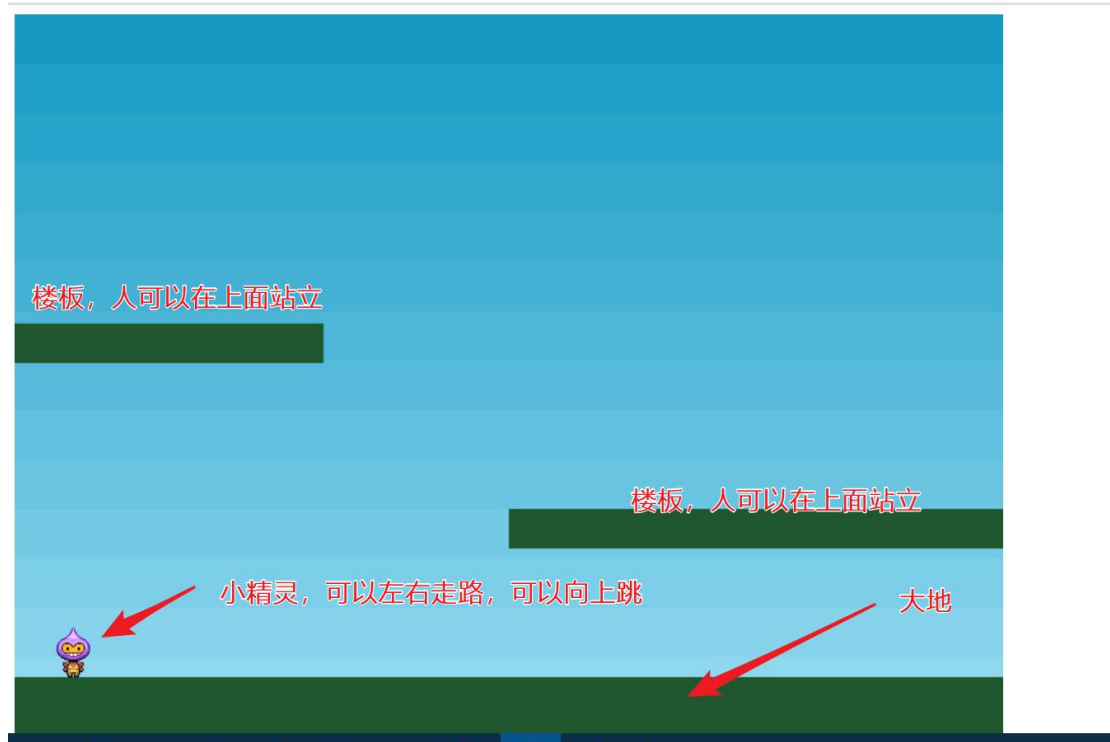
效果预览

https://zhengwei1949.github.io/making_your_first_h5_game/part9_1.html

案例来源

<http://phaser.io/tutorials/making-your-first-phaser-game>

需求说明：



吃掉星星可以得到积分

Phaser.js 优势

1、无须像白鹭引擎一样，要安装配置一堆的环境才能玩，只需要你的浏览器支持 canvas 标签就可以玩（当然，需要把代码放到服务器中去玩，否则存在跨域的问题）

第一步、搭建基本结构

```
1
2  var game = new Phaser.Game([
3      800, 600, Phaser.AUTO, '',
4      { preload: preload, create: create, update: update }
5  ]);
6
7  function preload() {
8  }
9
10 function create() {
11 }
12
13 function update() {
14 }
15
16 </script>
17
18 </body>
19 </html>
```

讲解：

1、我们使用 Phaser.Game 创建了 game 对象，我们后面的所有的东西都是要加载到 game 对象当中

2、800, 600 指的是我们的游戏的世界是 800*600

3、第三个参数的值可以是：Phaser.CANVAS, Phaser.WEBGL, Phaser.AUTO 推荐大家使用 Phaser.AUTO, 这样默认会使用 WebGL 的环境，如果不支持 WebGL 会退化为 canvas 的环境

第四个参数和页面中的 id 元素相关联，我们这里默认是空的，意思是会自动加载到 body 当中

3、最后一个参数里有三个函数, preload, create, update，这块并不是必须的，一般正式的游戏会创建很多的场景 state, 然后这些函数会在每个场景当中，我们这个例子比较简单，所以，这块就只有一个默认的场景，函数全写在这块

4、每个生命周期函数的执行次数

- (1) preload ---> create ---> update
- (2) preload,create 只执行一次, update 执行无数次
- (3) 我们可以在 preload,create,update 阶段打印整个世界的宽高

```

16
17 var game = new Phaser.Game(800, 600, Phaser.AUTO, '', { preload: pre
18
19 function preload() {
20     console.log(game.world.height);
21     console.log(game.world.width);
22     game.load.image('sky', 'assets/sky.png');
23     game.load.image('ground', 'assets/platform.png');
24     game.load.image('star', 'assets/star.png');
25     game.load.spritesheet('dude', 'assets/dude.png', 32, 48);
26
27 }
28
29 function create() {
30 }
31
32 function update() {

```

nc

top Filter Default le

⚠ [Deprecation] GainNode.gain.value setter smoothing is deprecated and will be removed in M64, around January 2018 instead if smoothing is needed. See <https://www.chromestat> for more details.

Phaser v2.0.1 - WebGL - WebAudio <http://phaser.io>

600

800

> |

```

27 function preload() {
28     game.debug.text('hello world', 32, 32);
29 }
30
31 function create() {

```

第二步、加载资源

```
17 var game = new Phaser.Game(800, 600, Phaser.AUTO, '', { preload: preload, create: create,
18
19 function preload() {
20
21     game.load.image('sky', 'assets/sky.png');
22     game.load.image('ground', 'assets/platform.png');
23     game.load.image('star', 'assets/star.png');
24     game.load.spritesheet('dude', 'assets/dude.png', 32, 48);
25
26 }
27
28 function create() {
29 }
```

讲解:

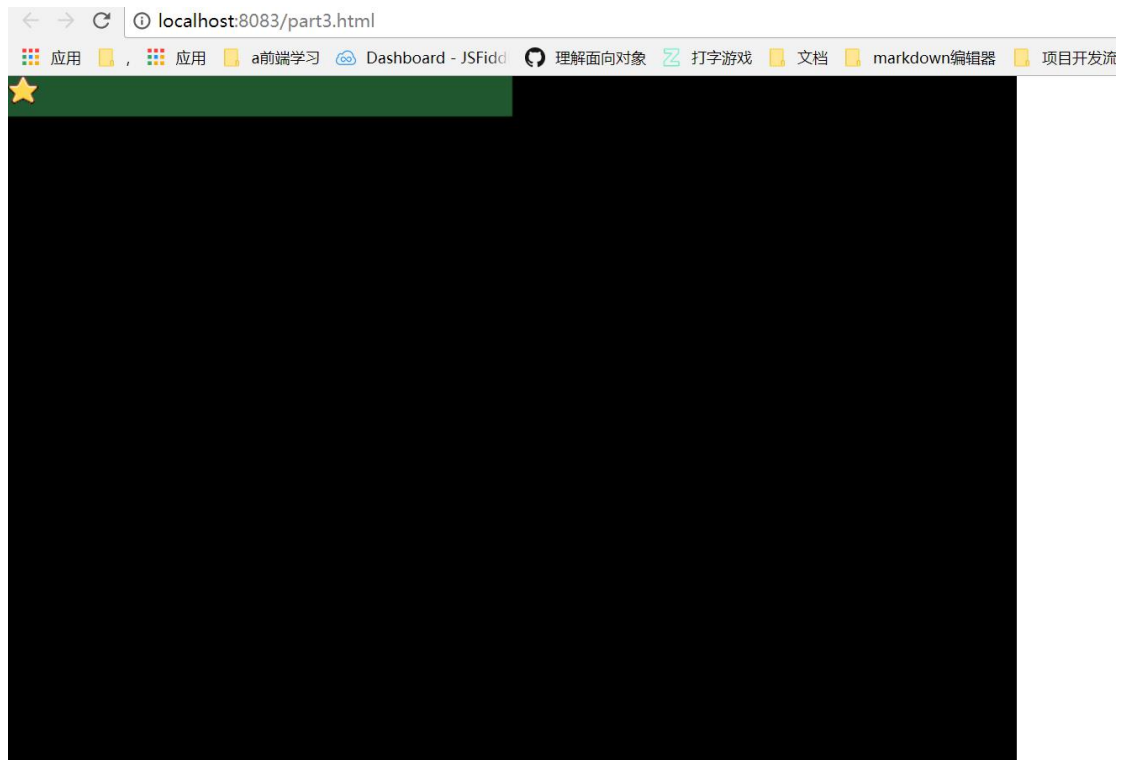
1、如果是精灵表(spriteSheet)，则要设置一下每个小精灵的宽度和高度，这块大家先这么做，后会明白为什么要这样写的

第三步、创建一个星星精灵和大地

这块是为了测试代码是否 ok,这两个小精灵后面会移除掉

```
23     game.load.image('star', 'assets/star.png');
24     game.load.spritesheet('dude', 'assets/dude.png', 32, 48);
25
26 }
27
28 function create() {
29     game.add.sprite(0, 0, 'ground');
30     game.add.sprite(0, 0, 'star');
31
32 }
33
34 function update() {
35 }
36
37 </script>
38
39 </body>
40 </html>
```

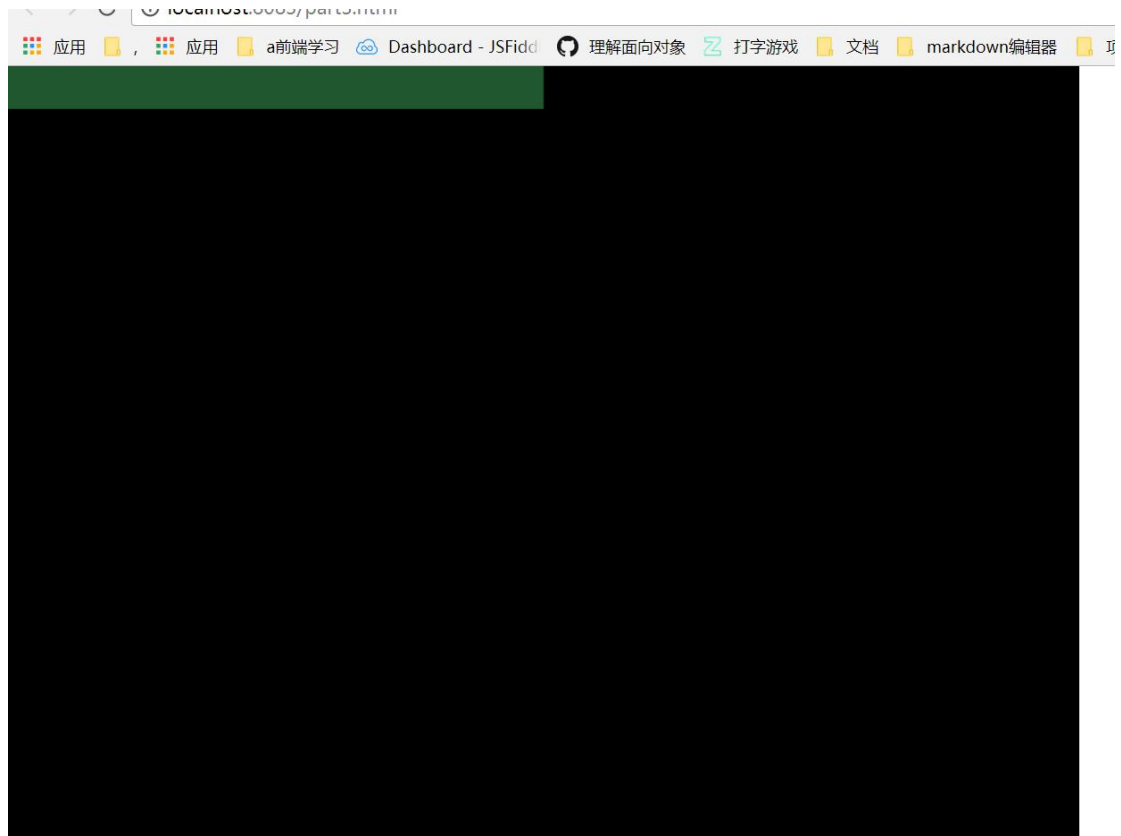
效果预览:



讲解：

1、精灵的顺序是后面的层级比较高，大家不要反着如下这样写，否则会看不到星星：

```
23     game.load.image('star', 'assets/star.png');
24     game.load.spritesheet('dude', 'assets/dude.p
25
26 }
27
28 function create() {
29     game.add.sprite(0, 0, 'star');
30     game.add.sprite(0, 0, 'ground');
31
32
33 }
34
35 function update() {
```



2、默认的坐标原点是左上角(0,0),当然这块是可以修改的（通过 camera，暂时我们用不上，不用管）

第四步、创建整个世界