



PixiJS 入门与实践

宁勃 20180531

1. PixiJS 是什么
2. PixiJS 能做什么
3. PixiJS 怎么用
4. PixiJS 实践

PixiJS v4 是什么

CSS 开启硬件加速: `translate3d/rotate3d/scale3d/skew3d/matrix3d/will-change`

Canvas 开启硬件加速: `var ctx = canvas.getContext('webGL')`

PixiJS是HTML5 创建引擎, 使用最快最灵活的2D WebGL 渲染。

优势: 渲染速度快, 灵活, API丰富。

PixiJS v4 能做什么

- Canvas动画: <http://go.163.com/2017/0805/neteaseCreative/index.html?code=071KEqZC0graMd2hKFXC0rrrZC0KEqZT&state=1>
- 滑屏长图: <http://go.163.com/2018/0120/taidu/index.html>
- 画中画: <http://s.auto.163.com/2017/1204/cadillac/index.html>
- 贴图拖拽: http://go.163.com/web/20180516_womai/index.html

PixiJS v4 常见名称

1. Application -> 初始化, 创建画布
2. Sprite -> 精灵, 基础
3. Texture -> 纹理, 精灵的基础
4. Container -> 类似DIV嵌套/数组
5. AnimatedSprite -> 序列帧
6. Loader -> 预加载函数
7. Ticker -> 每秒让画布重新渲染60次
8. Mask -> 遮罩
9. Graphics -> 基础图形

PixiJS v4 怎么做



PixiJS v4 画布 -> Application

Application 会自动创建一个Canvas;

```
let app = new PIXI.Application({  
    width: 256, // default: 800  
    height: 256, // default: 600  
    antialias: true, // default: false  
    transparent: false, // default: false  
    resolution: 1 // default: 1  
});
```

```
document.body.appendChild(app.view);
```

```
// app.stage 为画布的根容器
```

```
// 演示: https://codepen.io/ningbo/
```

PixiJS v4 怎么做



PixiJS v4 drawImage -> 精灵(Sprite)

Sprite -> Texture -> BaseTexture -> GPU

A texture stores the information that represents an image. All textures have a base texture

```
var base = new PIXI.BaseTexture(image);  
var texture = new PIXI.Texture(base);  
var sprite = new PIXI.Sprite(texture);  
Base.on( 'loaded' ,() => {  
    console.log( "image 载入成功" );  
});
```

// 纹理缓存

```
let texture = PIXI.utils.TextureCache["images/anySpriteImage.png"];  
let sprite = new PIXI.Sprite(texture);
```

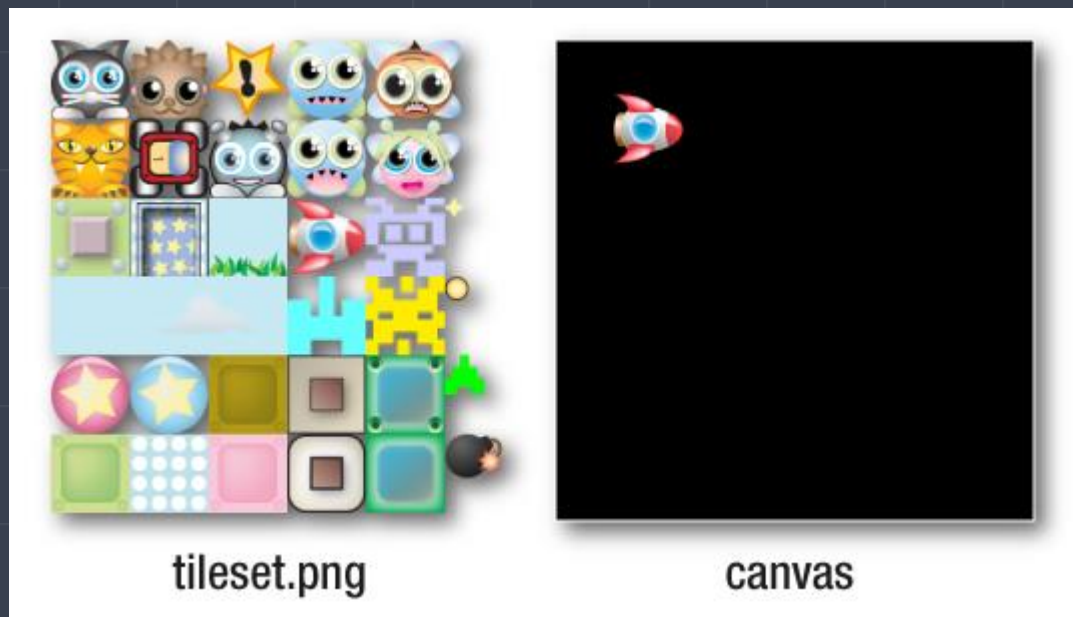
怎么加载图像并将它转化成纹理？答案是用Pixi已经构建好的loader对象。

```
PIXI.loader
.add( 'bunny' , ' https://pixijs.io/examples/required/assets/basics/bunny.png' )
.load()=>{
    var sprite = new PIXI.Sprite(
        PIXI.loader.resources[ 'bunny' ].texture;
    );
    app.stage.addChild(sprite);
};
```

PixiJS v4 loader加载的三种方法

1. 用一个单图像文件创建。
2. 用一个雪碧图来创建。雪碧图是一个放入了你游戏所需的所有图像的大图。
3. 从一个纹理贴图集中创建。（纹理贴图集就是用JSON定义了图像大小和位置的雪碧图）

PixiJS v4 雪碧图



// 加载

```
loader.add( "images/tileset.png" );  
let texture = TextureCache["images/tileset.png"];
```

// 定义矩形形状

```
let rectangle = new Rectangle(192, 128, 64, 64);
```

// 剪裁

```
texture.frame = rectangle;
```

// Create the sprite from the texture

```
let rocket = new Sprite(texture);
```

```
rocket.x = 32;
```

```
rocket.y = 32;
```

PixiJS v4 纹理图集

使用TexturePackerGUI 导出 Json 文件+雪碧图

Json: http://go.163.com/web/20180320_medspa/img/json/girl1.json

雪碧图: http://go.163.com/web/20180320_medspa/img/json/girl1.png

```
loader.add("girl","https://go.163.com/web/20180320_medspa/img/json/girl1.json")  
  .load(() => {init();});
```

```
let init = () =>{  
    let textures = loader.resources.girl.textures;  
    let sprite = new Sprite(textures["girl1.png"]);  
    app.stage.addChild(sprite);  
}
```

PixiJS v4 怎么做



PixiJS v4 requestanimationframe -> Ticker

ticker 1s更新60次

Sprite的基本属性：x,y,width,height,scale,rotation,alpha...

```
let ticker = new PIXI.ticker.Ticker();
```

```
ticker.add(()=>{
```

```
    sprite.rotation += 0.1;
```

```
});
```

```
ticker.start();
```

```
ticker.stop();
```

PixiJS v4 Container

// 可以理解为DIV / 数组

```
let container = new PIXI.Container();  
container.addChild(sprite);
```

// Container与ParticleContainer

ParticleContainer 的渲染速度比 Container 高 2~5 倍
ParticleContainer只能使用一个纹理，且禁止嵌套

PixiJS v4 Graphics+Mask

Graphics几何图形
Mask 遮罩

Mask既可以使用几何图形，也可以使用图片，图片中黑色与透明的部分均视为透明。

Container+Graphics+Mask组合使用: http://go.163.com/web/20180101_pixi_rect/index.html

PixiJS v4 常见问题

1. Canvas旋转后，点击位置会发生偏移，可以尝试用DOM浮层解决。

2. 纹理（Texture）只有在loader之后才可以读取宽高等信息。

3. 使用webGL toDataURL时需添加如下代码：

```
gl = canvas.getContext("experimental-webgl", {preserveDrawingBuffer: true});
```

4. 使用图片蒙版，黑色和透明均视为透明。

5. anchor与pivot会改变圆心的位置，可以在position加上位移恢复原来的位置。

感谢聆听