**宫廷H5-规范**

程序规范

1. **内存和性能**  
   **内存**

1. 代码定为4M以内（超过4m需要分包处理）

2. 资源最多50M（可以放在远程服务器上）[小游戏缓存限制为50M，实际有出入](<https://developers.weixin.qq.com/minigame/dev/tutorial/ability/file-system.html>)

**DrawCall**

在层级管理器中，从上往下顺序渲染，相邻两个结点同一个图集，drawcall可以合并，艺术字/图集交替不会合并drawcall。

单个精灵和系统字增加一个就会增加一个drawcall。  
（思考：可不可以和fairyGUI一样优化，修改单个精灵的渲染顺序）

参考优化

https://blog.csdn.net/zzx023/article/details/85319733

**降低结点复杂度**  
节点树越复杂刷新的成本越高（也就是节点树的深度尽量浅）。

**对象池**节点树越复杂刷新的成本越高

1. **数据存储**

1. 调用\*\*\*cc.sys.localStorage.setItem\*\*\* 存储数据

2. 调用\*\*\*cc.sys.localStorage.getItem\*\*\* 获取存储数据

3. 若存储的值为对象序列化成json格式再进行存储

1. **加载和释放**

**场景加载**

1. 使用preloadScene可以实现场景的预加载功能

2. 在跳转场景的时候，先跳转一个Loading的场景，再跳转到目标场景，防止跳转目标场景时间过长，导致用户异常操作

**资源加载**

1. 动态图片资源加载

2. 声音资源加载（可以直接拖进场景里面）

3. prefab（可以直接拖进场景里面）

4. animation

``` js

//合图资源加载

cc.loader.loadRes(path, cc.SpriteAtlas, function (err, atlas) {

if(err){

console.error("loadAtlasRes err in AtlasResLoader, err =", err);

return;

}

//loaded success

});

//声音资源加载

cc.loader.loadRes(soundUrl,cc.AudioClip,(err,soundClip)=>{

if(!err){

cc.audioEngine.playEffect(soundClip,false,1);

}

else{

console.error("err : " + err);

}

})

//加载某一路径下的资源

// 加载 Prefab

cc.loader.loadRes("test assets/prefab", function (err, prefab) {

var newNode = cc.instantiate(prefab);

cc.director.getScene().addChild(newNode);

});

// 加载 AnimationClip

var self = this;

cc.loader.loadRes("test assets/anim", function (err, clip) {

self.node.getComponent(cc.Animation).addClip(clip, "anim");

});

```

**资源释放**

1. prefab

2. 动态资源

3. 静态资源

4. animation

``` js

//设置资源的自动释放

cc.loader.setAutoRelease(cc.SpriteFrame, true);

cc.loader.setAutoRelease(cc.Texture2D, true);

cc.loader.setAutoRelease(cc.Prefab, true);

cc.loader.setAutoReleaseRecursively(cc.SpriteFrame, true);

cc.loader.setAutoReleaseRecursively(cc.Texture2D, true);

cc.loader.setAutoReleaseRecursively(cc.Prefab, true);

//必要的时候可以手动释放

// 直接释放某个贴图

cc.loader.release(texture);

// 释放一个 prefab 以及所有它依赖的资源

var deps = cc.loader.getDependsRecursively('prefabs/sample');

cc.loader.release(deps);

//释放某一路径下的资源

cc.loader.releaseRes("test assets/image", cc.SpriteFrame);

cc.loader.releaseRes("test assets/anim");

1. **代码规范**

1. 使用驼峰命名法

2. 使用mvc开发模式