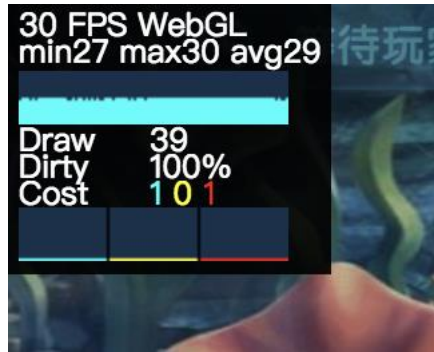


Egret 性能优化方案

1. FPS 优化



- a) 蓝色：cpu 消耗系数，手机上不能高于 10（0-10）
- b) 黄色：事件消耗系数，手机上不能高于 5（0-5）
- c) 红色：gpu 消耗系数，手机上不能高于 10（0-10）
- d) 如果有超过范围的，必须想办法优化，哪怕是偶尔的跳动。
 - i. 蓝色：针对代码逻辑优化（运算量，复杂度，耦合度，等等）
 - 1. 不适用的定时器，一定要回收
 - 2. 尽可能重用对象，建立对象池，而不创建对象并对其执行垃圾回收
 - 3. 多次调用类属性时，避免直接使用 `this.att`，建立局部变量赋值

4. 减少不必要的方法类的引用
5. 减少显示对象的旋转缩放
6. 删除对象的所有引用确保被垃圾回收器回收
7. 尽量不要在使用方法内创建函数，一个函数内不要
嵌套另一个函数的定义
8. 尽量使用 `let` 代替 `var` 避免闭包导致的 BUG
9. 将 `this` 赋值给另一个临时变量 `self`，`let self = this`
10. 不要在声明的时候初始化对象，需要赋值基本类
型。可选择使用前初始化
11. 初始化数组 `=[]` 比 `new` 效率高，`= {}` 比 `new` 效
率高
12. `exml` 文件注意通过可视化编辑会产生小数，去掉小
数
13. 少用 `mask`，特别是 `Rectangle` 每回都要去计算消耗
性能较大
14. 配置文件尽量使用 `key : value` 的方式，直接通过

key 取值减少查询

ii. 黄色：针对事件的调用和添加进行优化（主要在于频繁添加事件和派发事件）

1. 不需要的事件侦听器，一定要移除
2. 不需要的触摸事件，请关闭交互
3. 合理使用 `dispatchEvents` 函数
4. `Event.ENTER_FRAME` 数量控制

iii. 红色：针对渲染优化。

1. 减少 `alpha` 的混合（特别是动画和频繁设置 `alpha` 值）
2. 使用 `SpriteSheet` 合并的图片尺寸要优于单张图片的总尺寸，尤其是带透明通道的
3. 显示移除不需要的面板，不用的尽量 `removeChild` 而不是设置 `visible` 或者设置 `alpha`
4. 静态的 UI 使用建议使用 `cacheAsBitmap` 减少重绘次数，动态的千万不要使用该方法

5. TextField 减少对于描边 (stroke) 的使用，用图片代替，因为描边是需要先通过计算边界渲染

2. 特效性能优化

- a) 多个 Effect 同界面播放的时候，请误直接调用 play 放，使用界面帧事件，统一调用 update 方法来播放
- b) 播放时间不为-1 (不为循环播放) 的特效，一定要重利用，不能直接 new，改变当前播放时间和特效总时长即可重利用

3. 界面性能优化

- a) 不需要的帧事件，请不要设置 isFrame
- b) 尽量把实例化操作放到 onCreate，避免重复实例化
- c) 尽量少使用 layer 作为一个组件
- d) 使用 egret.Tween 动画的，特别是循环的动画，一定要在 onExit 中移除，避免内存泄漏
- e) 不需要的 egret.Tween，一定要 egret.Tween.removeTweens()

4. 计算、重绘频率优化

- a) 减少刷新频率，尽量不使用 60 帧的运算频率，默认 30 帧，
减少运算从而减低发热。