



**QCon** 全球软件开发大会  
INTERNATIONAL SOFTWARE  
DEVELOPMENT CONFERENCE

BEIJING 2017

# 骨骼动画实践

李振文 (Viktorli) 腾讯

- 骨骼动画介绍
- 宠物项目实践技术点
- 遇到的问题点及解决方案
- 性能优化

# 帧动画



# 骨骼动画



# 骨骼动画

- 资源体积更小
- 多角色可共用一套动作数据
- 动作可以自由组合
- 动画更逼真
- 对处理器的性能要求更高

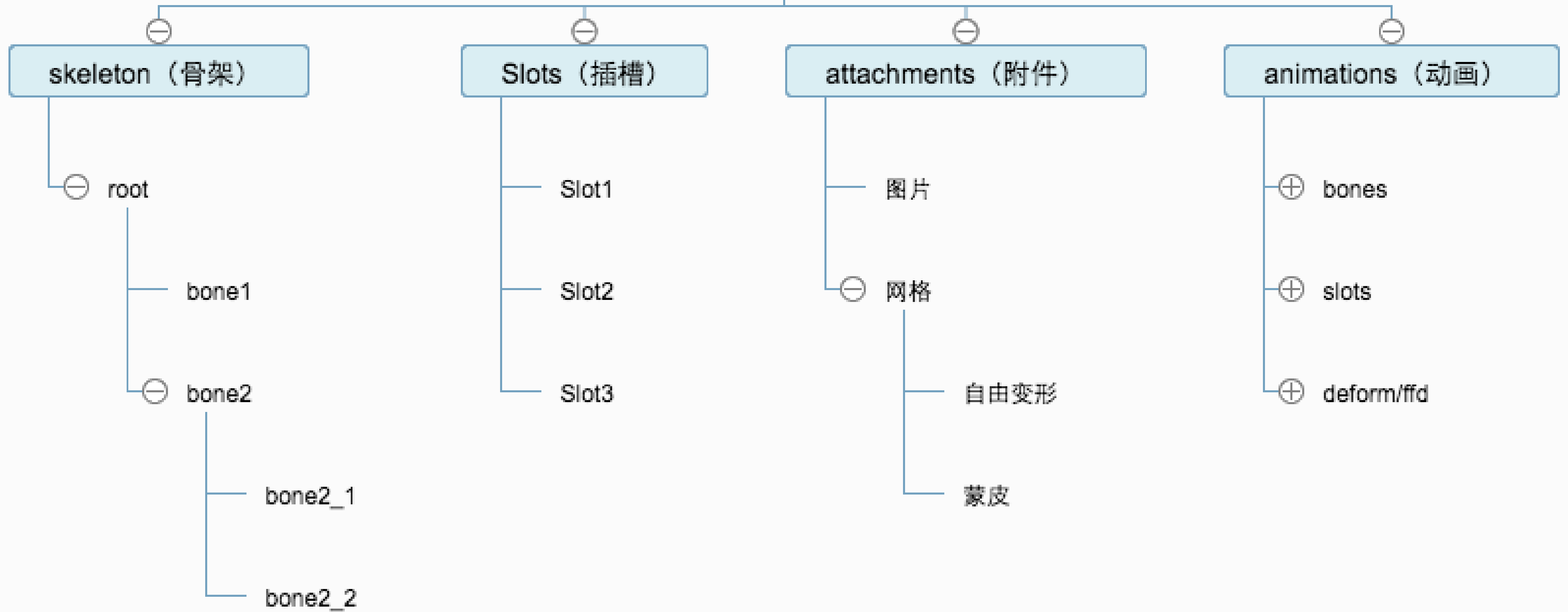




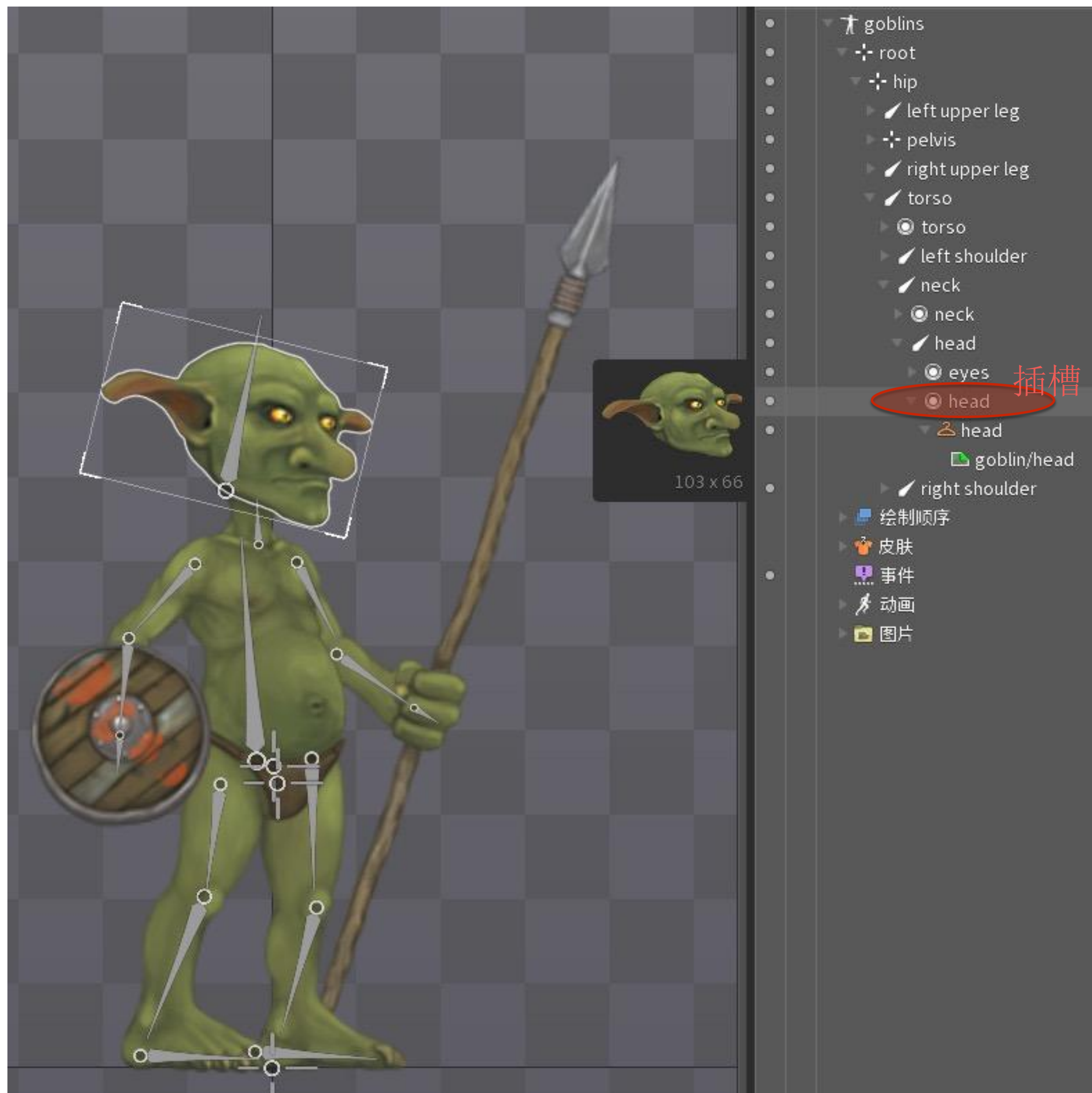




# 骨骼动画







## ◆ 骨架

- 躯干

- ▶ 左臂
- ▶ 右臂
- ▶ 脖子

- 左腿

- ▶ 左大腿
- ▶ 左小腿

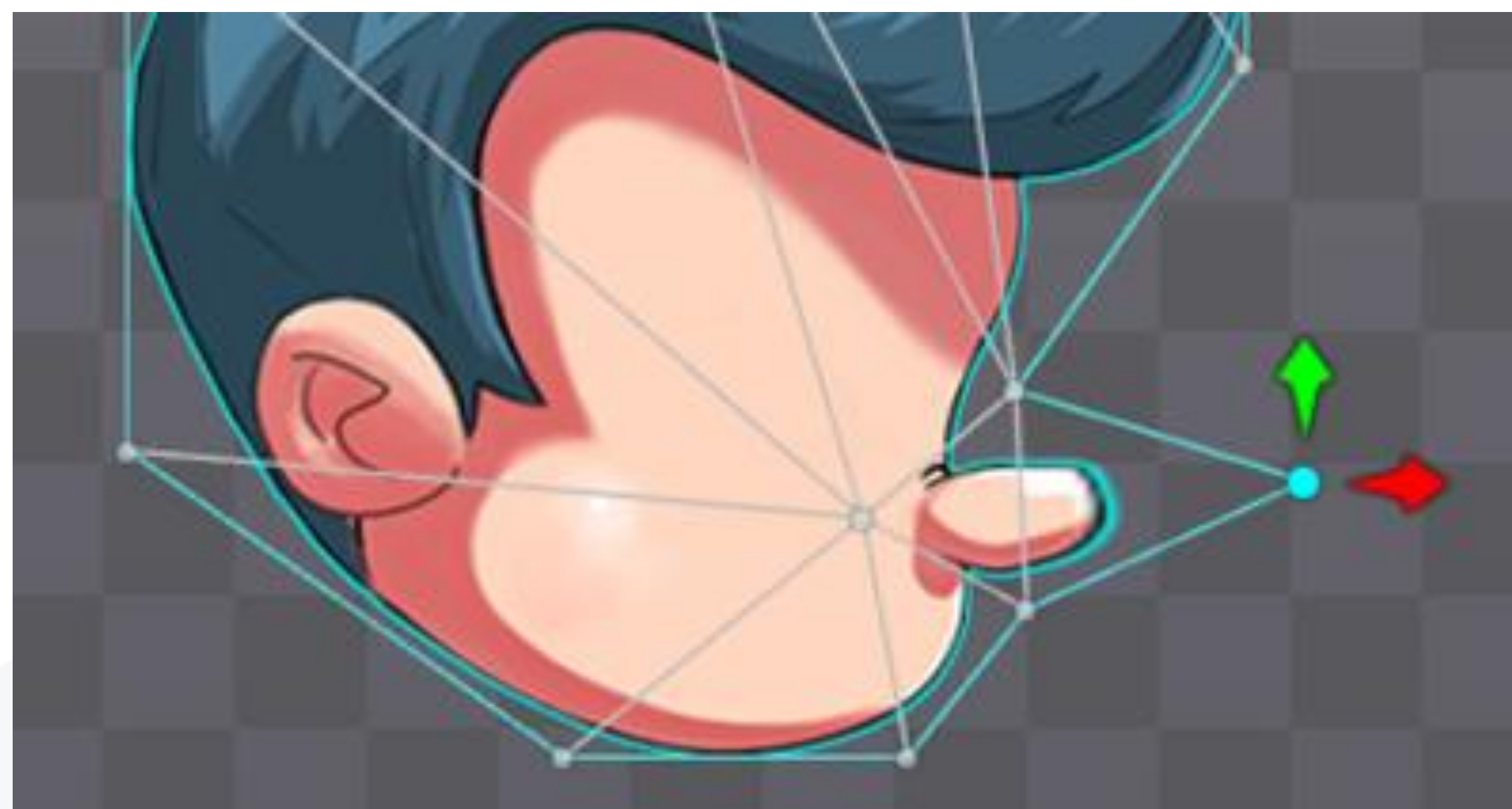
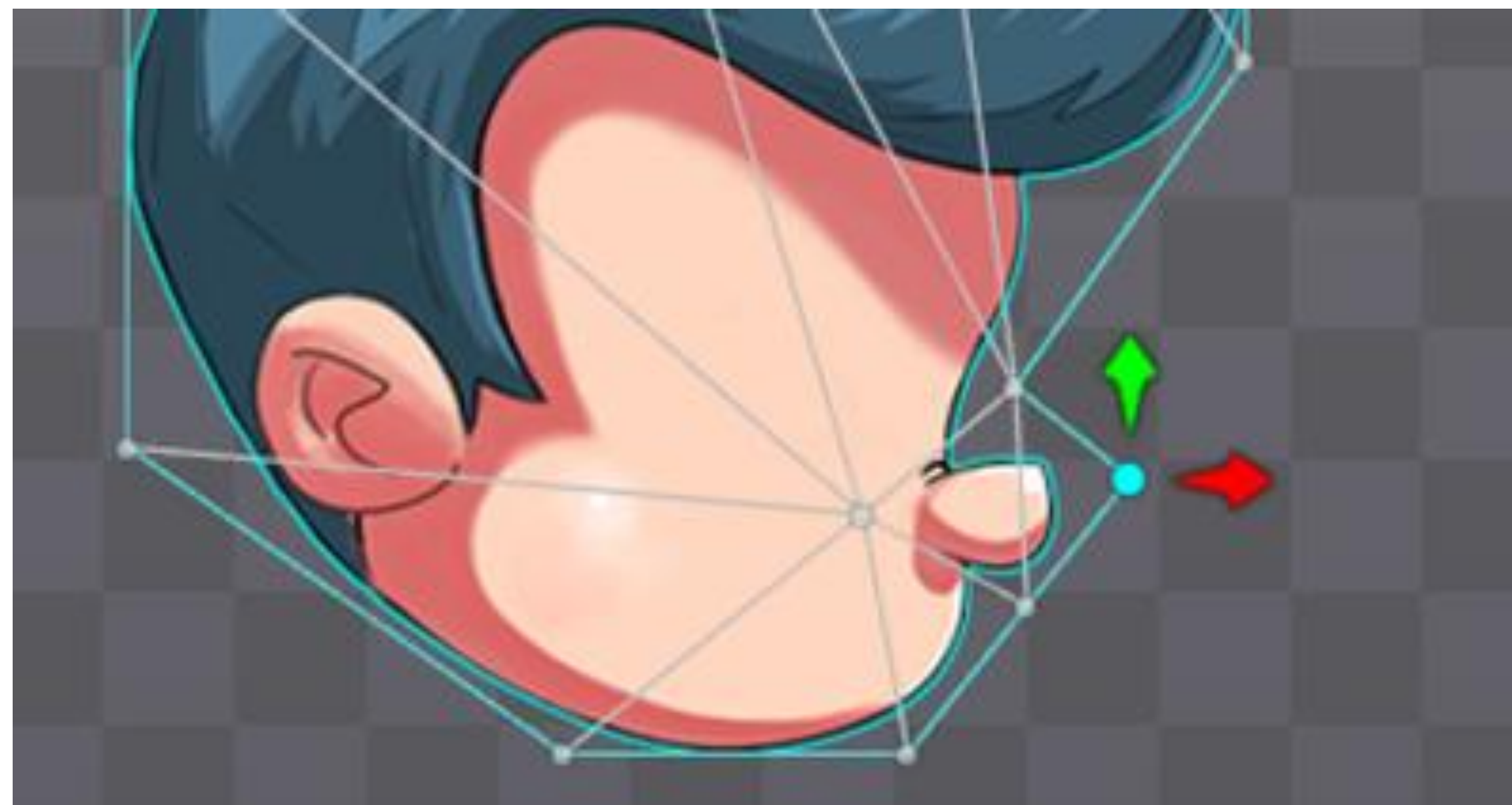
- 右腿

- ▶ 右小腿
- ▶ 右大腿

- 盆骨

- ▶ 腰带
- ▶ 裤衩













付费

运行库众多

功能更多

普及程度高，可选的设计公司更多

免费，国产！

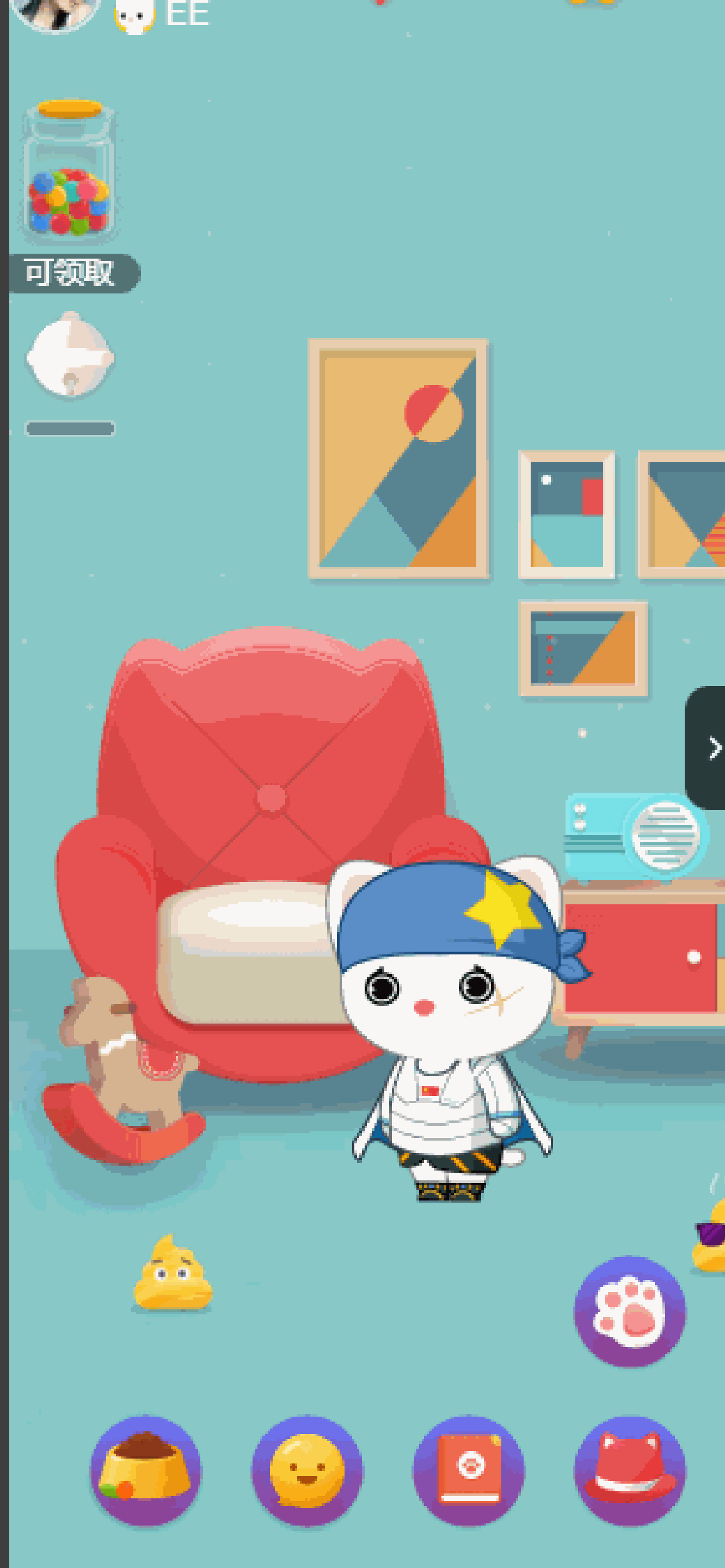
运行库较少

功能相对少，但作者不断完善中

可以与作者直接交流提需求！



SPINE基础库	性能	webgl	体积	API/文档	活跃度	总结
PIXI	60FPS		293KB+63.9 KB			轻量 活跃度及API都不错 与SPINE团队合作较紧密
Three.js	60FPS		499KB+70KB			支持3d, 库偏大 项目仅 需要2d
Cocos2d-html5						太旧, 无人维护状态
PlayCanvas	60FPS		510KB+50.7 KB			封装得很易用 API太少 文档和示例缺失

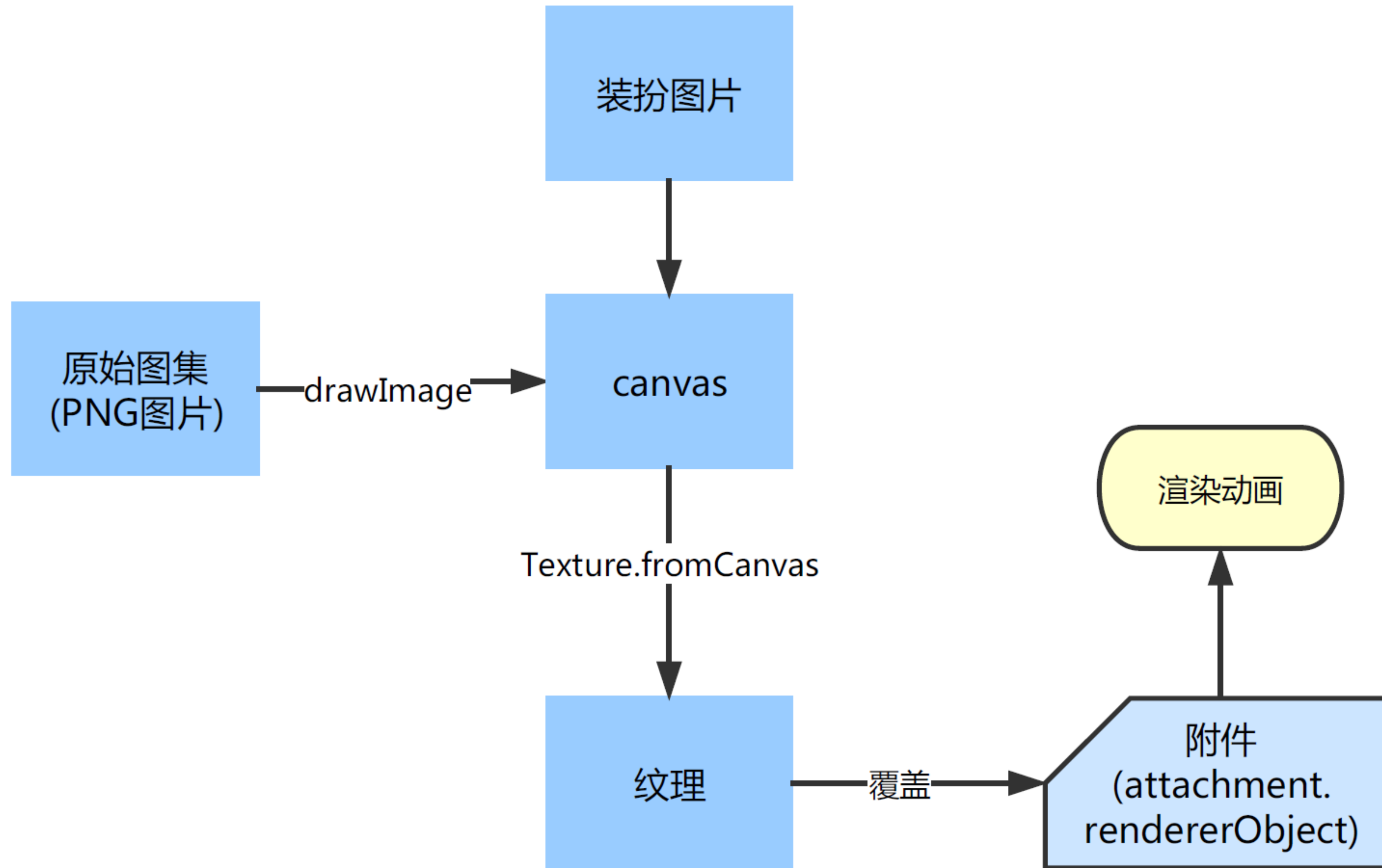


- 1  
Lv.21  
GuangTa
- 2  
Lv.18  
龙妹妹
- 3  
Lv.14  
GuangT
- > 4  
Lv.14  
ZX□
- 5  
Lv.11  
蓝色忧伤
- 6  
Lv.18  
龙妹妹

# 技术实现： 实时换装



# 技术实现：实时换装





# 技术实现： 实时换装

//清除旧衣服在canvas上的位置

```
canvas.clearRect(0, 0, width, height);
```

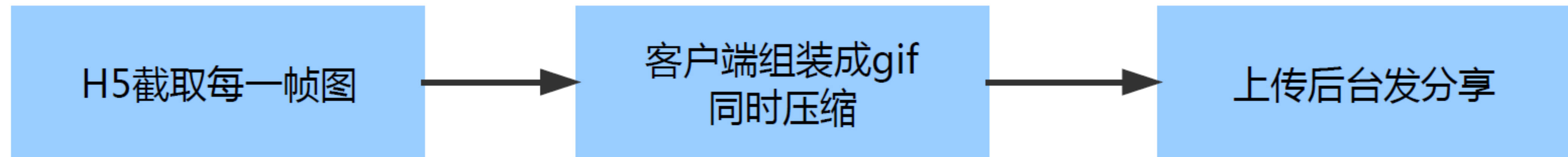
//将新衣服画在canvas上面

```
canvas.drawImage(image, 0, 0, width, height);
```

//刷新纹理

```
rendererObject.update();
```

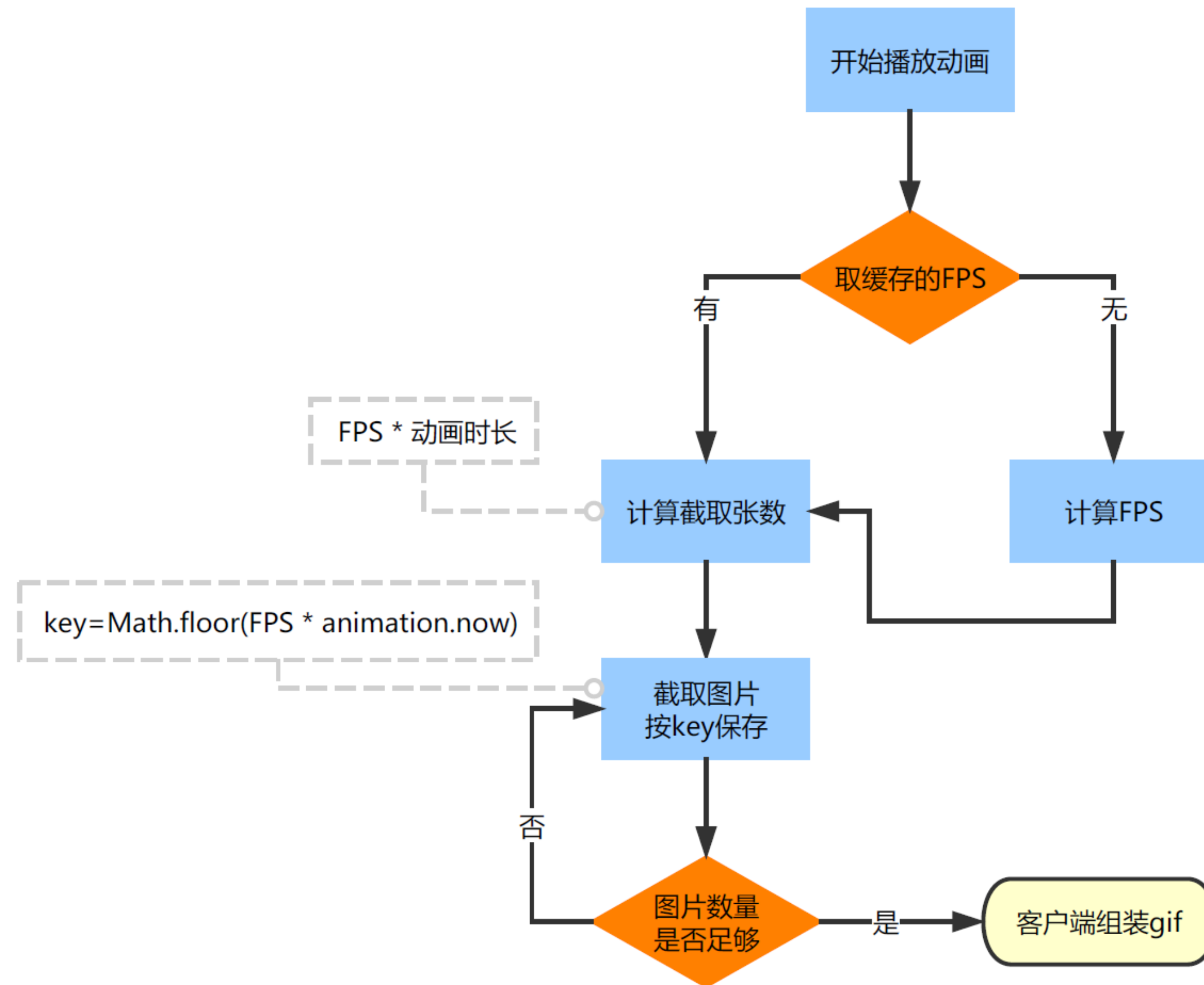
# 技术实现： 分享GIF



# 技术实现： 分享GIF



# 技术实现： 分享GIF





# 技术实现： 分享GIF



# 问题：展示错乱



# 问题：展示错乱

```
...
if (type === spine.AttachmentType.skinnedmesh
    || type === spine.AttachmentType.mesh
    || type === spine.AttachmentType.linkedmesh)
{
    if (slot.currentSprite) {
        slot.currentSprite.visible = false;
        slot.currentSprite = undefined;
        slot.currentSpriteName = undefined;
    }
}

....
}
```

# 问题：蒙皮类动画闪烁





# 问题：mesh类动画闪烁

```
PIXI.glCore.VertexArrayObject.FORCE_NATIVE = true;
```



# 问题：mesh类动画闪烁

```
/**
 * Helper class to work with WebGL VertexArrayObjects (vaos)
 * Only works if WebGL extensions are enabled (they usually are)
 *
 * @class
 * @memberof PIXI.glCore
 * @param gl {WebGLRenderingContext} The current WebGL rendering context
 */
function VertexArrayObject(gl, state)
{
    this.nativeVaoExtension = null;

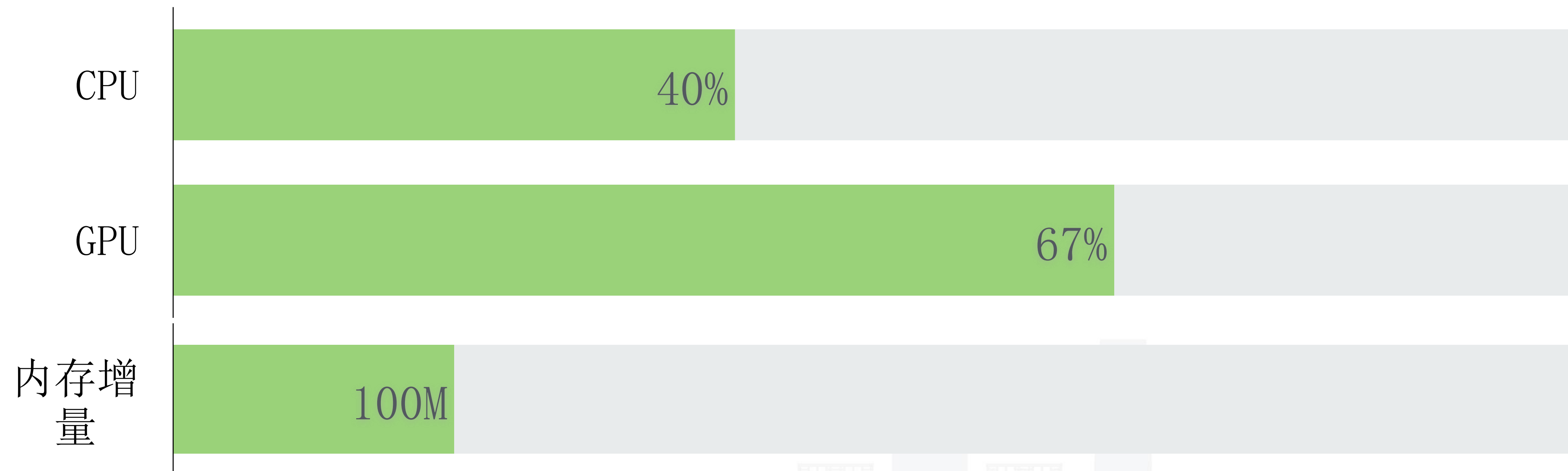
    if(!VertexArrayObject.FORCE_NATIVE)
    {
        this.nativeVaoExtension = gl.getExtension('OES_vertex_array_object') ||
            gl.getExtension('MOZ_OES_vertex_array_object') ||
            gl.getExtension('WEBKIT_OES_vertex_array_object');
    }
}
```

# 小结

- 熟读源码
- 与作者交流

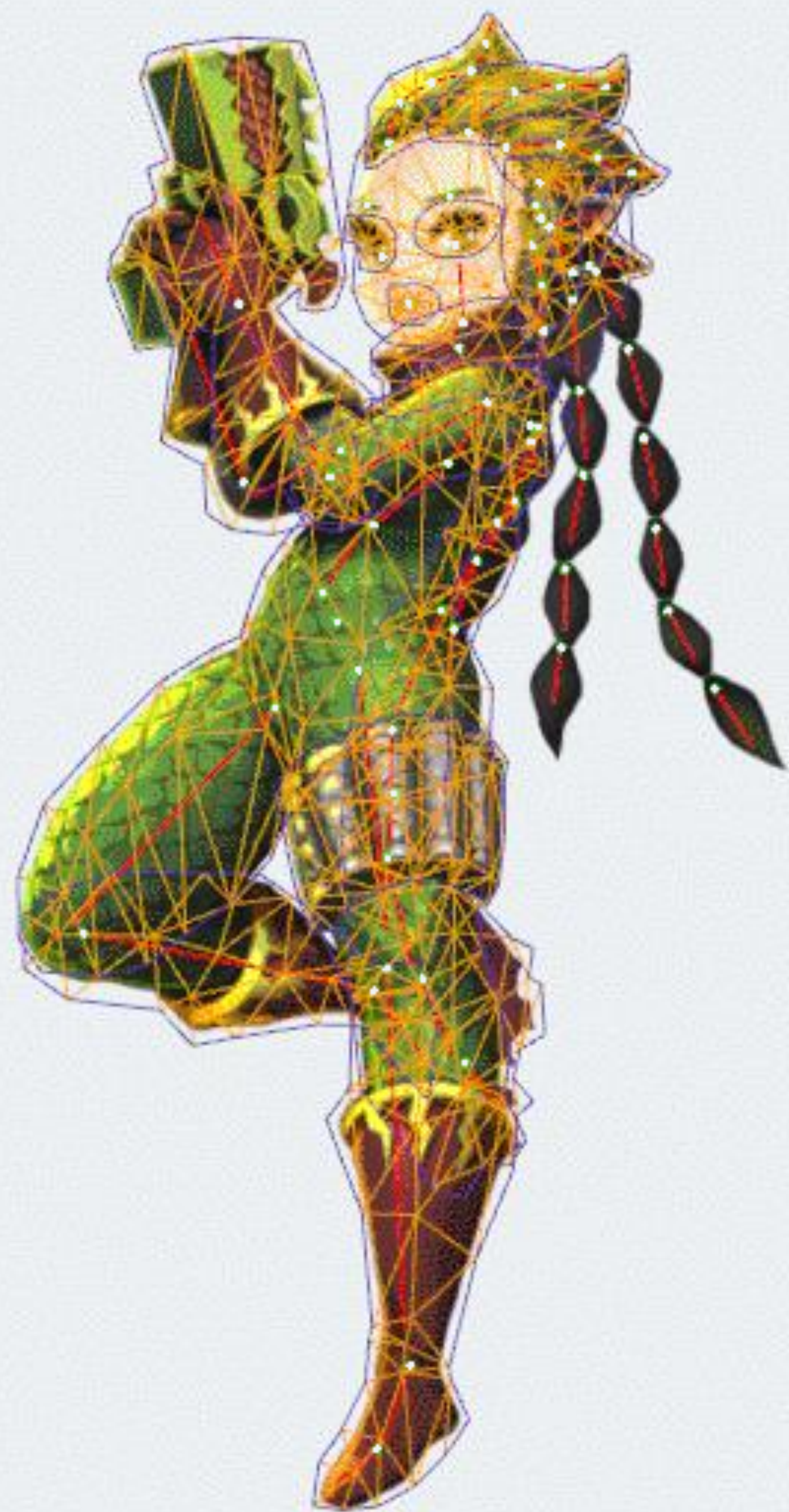


# 性能调优



耗电  
卡顿  
发热  
崩溃



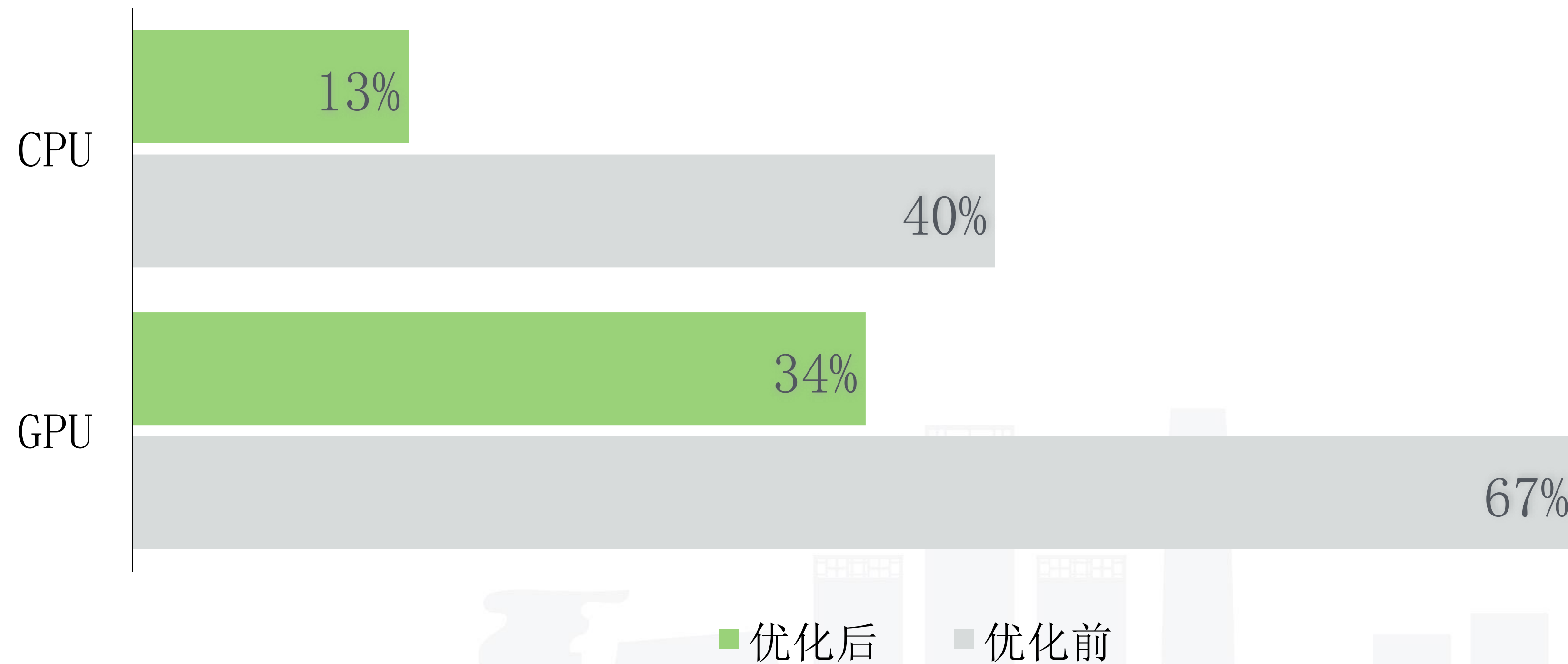


# 性能调优：CPU/GPU占用

- 减少每一帧运动的骨骼及网格数量
- 将待机动作（idle）改成隔几秒动一次
- APP切换到后台时停止动画



# 性能调优：CPU/GPU 占用





# 性能调优：减少内存占用

图片

单倍图

37M

双倍图

38M

精简的动作数据

14M

完整的动作数据

49M

# 性能调优：减少内存占用

- 暴露私有接口spineJsonParser

```
var spineJsonParser = new spine.SkeletonJsonParser(new spine.AtlasAttachmentParser(spineAtlasParser));  
var skeletonData = spineJsonParser.readSkeletonData(resource.data);  
  
skeletonData.spineJsonParser = spineJsonParser;  
resource.spineData = skeletonData;  
resource.spineAtlas = spineAtlas;  
if (atlasParser.enableCaching) {  
    atlasParser.AnimationCache[resource.name] = resource.spineData;  
}
```



# 性能调优：减少内存占用

文件大小

精简动作数据(4个)

201KB

完整动作数据(70个)

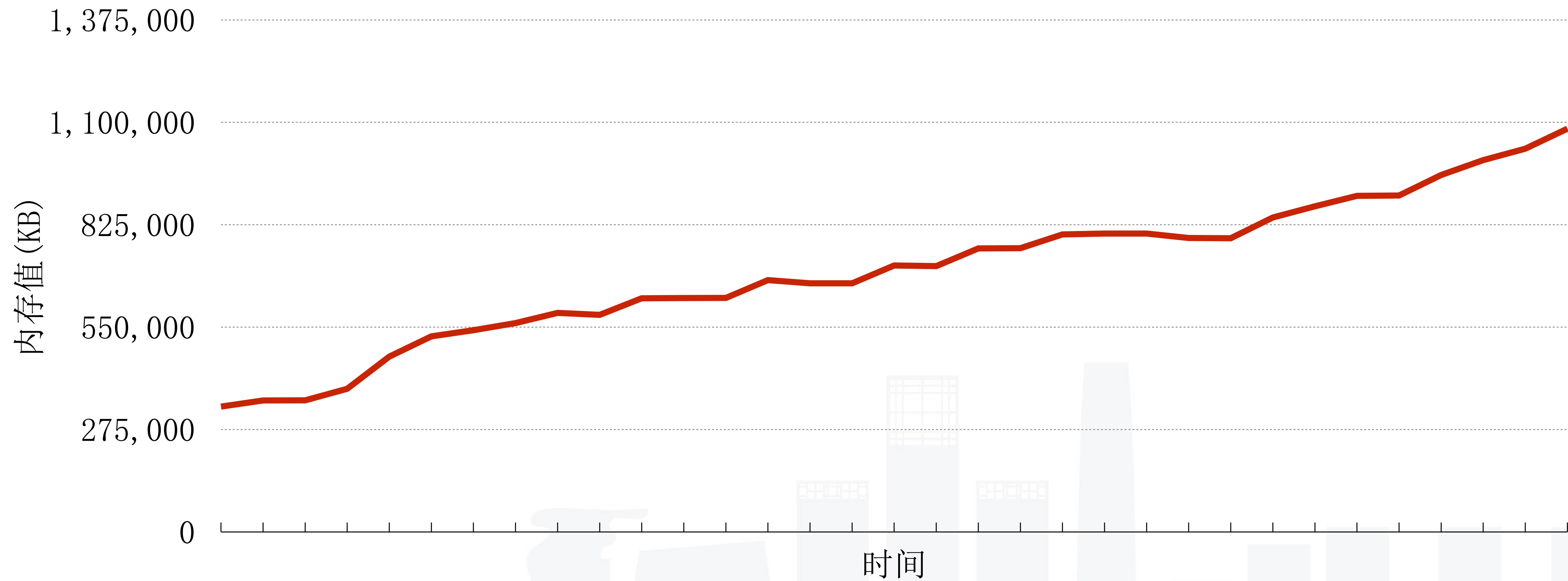
3.5MB

# 性能调优：减少内存占用

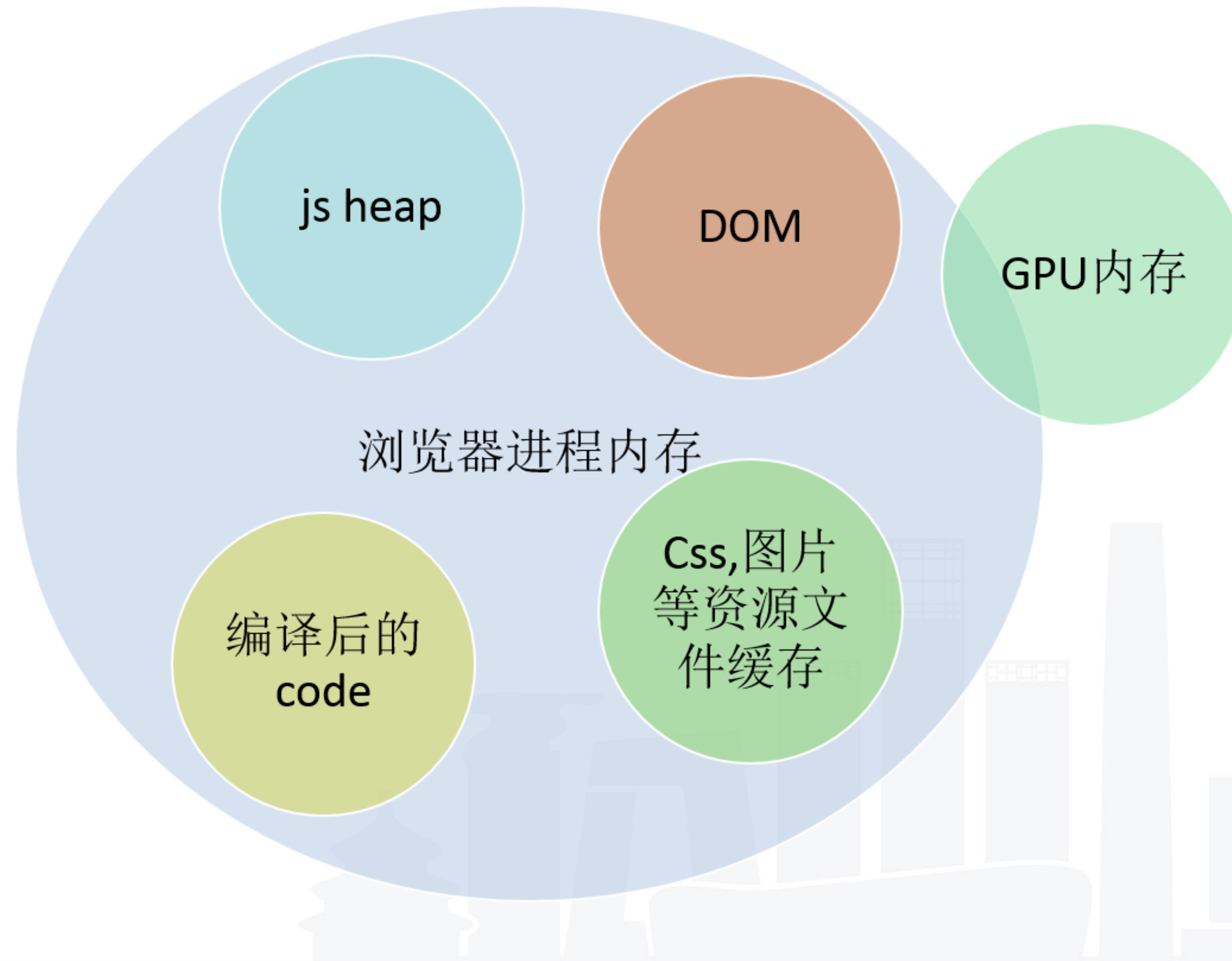
49M → 18M



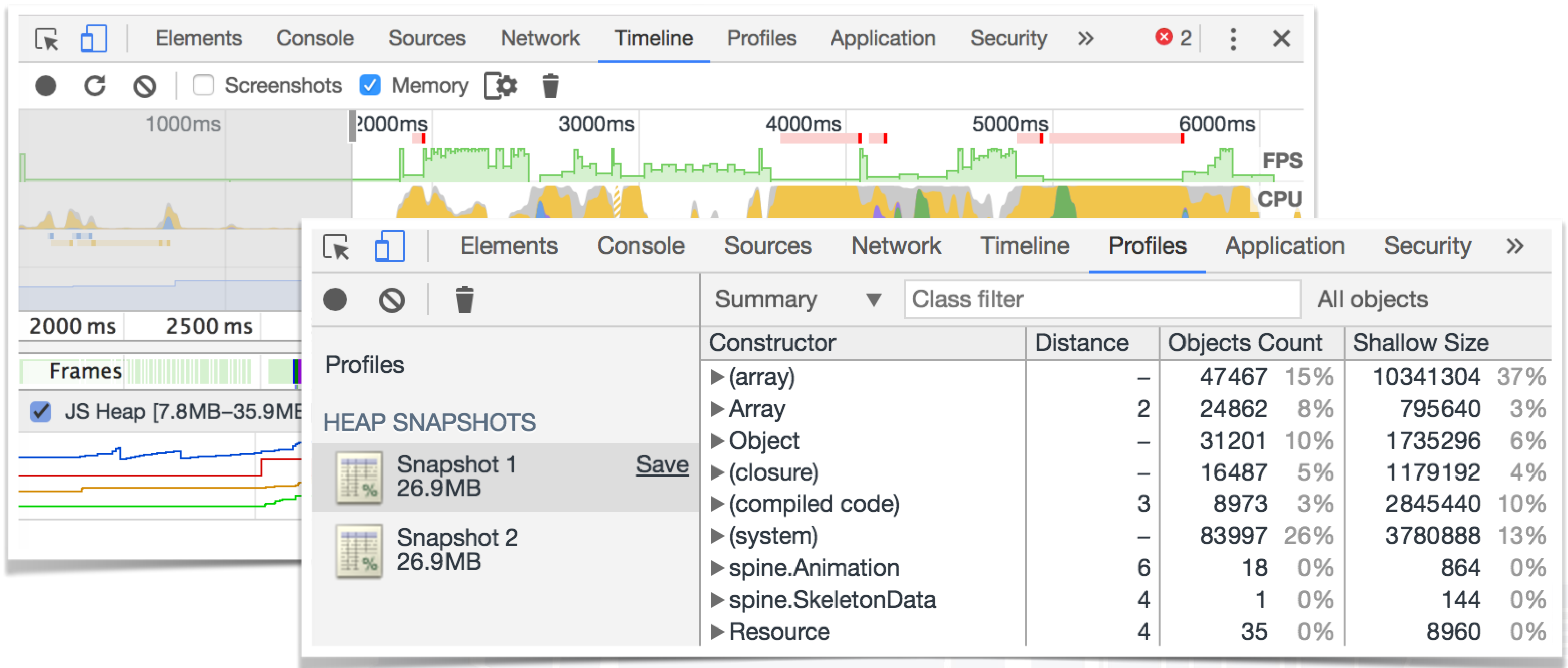
# 性能调优：切换好友内存泄漏



# 性能调优：切换好友内存泄漏



# 性能调优：切换好友内存泄漏





# 性能调优：切换好友内存泄漏

任务管理器 - Google Chrome

任务	内存	CPU	网络	进程 ID	GPU内存	CSS 缓存	图片缓存	JavaScript 使用的内存 ▼
• KM ...	91,132K	0.0	0	8112	9,279K	79.0 K ( 实际大小...	78,373 K ( 实际大...	27,688 K ( 实际大小为 17,802 K )
• ...	13,940K	0.0	0	5936	-	0 K ( 实际大小为 ...	3.3 K ( 实际大小...	7,516 K ( 实际大小为 5,730 K )
• ...	8,268K	0.0	0	5144	-	-	-	4,956 K ( 实际大小为 3,256 K )
• ...	10,744K	0.0	0	9964	-	0 K ( 实际大小为 ...	0 K ( 实际大小为 ...	4,956 K ( 实际大小为 3,184 K )
• ...	11,140K	0.0	0	984	-	0 K ( 实际大小为 ...	5.3 K ( 实际大小...	4,444 K ( 实际大小为 3,370 K )
• ...	10,288K	0.0	0	8236	-	0 K ( 实际大小为 ...	3.3 K ( 实际大小...	4,444 K ( 实际大小为 3,187 K )
• ...	8,208K	0.0	0	8864	-	0 K ( 实际大小为 ...	0 K ( 实际大小为 ...	3,932 K ( 实际大小为 2,170 K )
• ...	6,504K	0.0	0	7512	-	0 K ( 实际大小为 ...	0 K ( 实际大小为 ...	1,884 K ( 实际大小为 763 K )
• ...	78,724K	5.5	0	4636	26,730K	-	-	0 K ( 实际大小为 0 K )
• G...	30,816K	0.8	0	4884	36,009K	-	-	-

结束进程

# 性能调优： 切换好友内存泄漏

切换12个好友

JS内存增长

18M

GPU内存增长

90M

# 性能调优： 切换好友内存泄漏

- ◆ 同一种宠物模型对象复用，切换好友宠物就相当于换装

## JS内存增量优化



# 性能调优：切换好友内存泄漏

- ◆ 换装纹理复用
- ◆ 主动释放不再使用的纹理

```
page.rendererObject.dispose(true)
```

## GPU内存增量优化





# 性能调优： 切换好友内存泄漏

- ◆ 从排行榜切换回来之后，销毁好友的宠物数据，回收内存

```
pet.stage.destroy(true)
```



# 回顾

- 骨骼动画基础知识点
- 编辑器选择、开发引擎的选择
- 技术实现：实时换装、GIF截图
- 问题点：素材错位、蒙皮类动画闪烁
- 性能优化：CPU/GPU/内存的优化

# 总结

- 减少待机动画频率，降低CPU/GPU压力
- 动作及素材文件做成按需加载，减少内存占用，也能提高访问速度
- 模型/纹理尽可能地复用，减少内存占用
- 使用临时方案或者规范来解决紧急问题，但需要深究最佳方案，避免背负技术债
- 阅读源码、与原作者交流能帮助我们选择更好的方案解决问题
- 发布标准的制定，使用参照物对比

谢 谢





邮箱: [viktorli@qq.com](mailto:viktorli@qq.com)

欢迎索要PPT/示例、推荐简历