目的：

测试在高、中、低端机型上，特效层数的最大值（通过控制特效播放数量来稳定性能）。

特效分层功能：

通过给每个特效配置特效的优先级，根据人物关系，和特效优先级来确定特效的是否播放或者显示某个特效。

测试方法：

1. 分别在高、中、低端机器上使用10个不同角色，不停释放技能。
2. 从0层开始，向上设置特效的最大层数，间隔10层设置一次。
3. 收集相关参数，以DrawCall为指标，记录Camera.Render，meshSkinning.render ，culling，Particl.update，Particle.endupdateall。
4. 特效相关的CPU消耗为Camera.Render- meshSkinning.render- culling+ Particl.update+ Particle.endupdateall 因为Camera.Render中包含了meshSkinning.render和culling，而且这两个数据是一个动态数据，会影响到性能数据的统计，所以把这两个值从Camera.Render的总值中减去。
5. 设置层数时，记录4个drawcall的峰值，记录DrawCall数以及第4点提到的数据，根据第4点给出的公式算出特效的CPU消耗。
6. 设置的n层与0层的CPU消耗平均值差值接近10ms，则认为该机型适合的最大层数为n
7. 该测试屏蔽了伤害数字和血条，去除特效外的性能影响。
8. 由于很多buf类特效美术还没有做特效分层，所以此次测试使用固定的几个英雄和对应固定技能（做过特效分层的技能）。

数据记录（CR：Camera.Render，MSR：meshSkinning.render，Culling，PU：Particl.update，PE：Particle.endupdateall）

1. Vivo Y51A（低端机）：

0层：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DrawCall | CR(ms) | MSR(ms) | Culling(ms) | PU(ms) | PE(ms) | CPU(ms) |
| 167 | 19.66 | -4.81 | -1.91 | 0.11 | 0.06 | 13.11 |
| 169 | 19.79 | -4.76 | -1.65 | 0.09 | 0.05 | 13.52 |
| 168 | 20.8 | -4.38 | -1.69 | 0.14 | 0.04 | 14.91 |
| 167 | 22 | -5.36 | -1.97 | 0.08 | 0.03 | 14.78 |

均值：14.08

特效消耗：

30层：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DrawCall | CR(ms) | MSR(ms) | Culling(ms) | PU(ms) | PE(ms) | CPU(ms) |
| 223 | 28.5 | -5.13 | -6.37 | 0.92 | 1.52 | 19.44 |
| 218 | 25.6 | -3.64 | -3.22 | 1.46 | 2.12 | 22.32 |
| 214 | 39.06 | -12.3 | -8.36 | 0.69 | 1.16 | 20.25 |
| 222 | 24.53 | -4.48 | -3.95 | 0.71 | 1.67 | 18.48 |

均值：20.12

特效消耗：20.12-14.08=6.04ms

40层：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DrawCall | CR(ms) | MSR(ms) | Culling(ms) | PU(ms) | PE(ms) | CPU(ms) |
| 243 | 29.75 | -5.87 | -4.52 | 1.01 | 1.32 | 21.69 |
| 217 | 27.7 | -4 | -3.11 | 1.13 | 2.4 | 24.12 |
| 258 | 28.82 | -3.85 | -4.43 | 0.87 | 2.35 | 23.76 |
| 236 | 25.5 | -3.6 | -3.51 | 0.93 | 3.73 | 23.05 |

均值：23.16

特效消耗：23.16-14.08=9.08ms

50层：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DrawCall | CR(ms) | MSR(ms) | Culling(ms) | PU(ms) | PE(ms) | CPU(ms) |
| 271 | 30.51 | -4.03 | -5.62 | 1.09 | 3.58 | 25.53 |
| 274 | 27.92 | -3.65 | -4.91 | 2.36 | 2.67 | 24.39 |
| 257 | 28.44 | -4.77 | -5.15 | 1.43 | 3.74 | 23.69 |
| 261 | 30.31 | -4.59 | -4.61 | 1.23 | 2.29 | 24.63 |

均值：24.56

特效消耗：24.56-14.08=10.48ms

60层：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DrawCall | CR(ms) | MSR(ms) | Culling(ms) | PU(ms) | PE(ms) | CPU(ms) |
| 280 | 31.46 | -3.66 | -5.05 | 1.22 | 4.39 | 28.36 |
| 279 | 30.57 | -3.76 | -4.58 | 1.97 | 2.82 | 27.02 |
| 269 | 29.08 | -3.74 | -5.16 | 2.04 | 3.24 | 25.46 |
| 274 | 31.13 | -3.6 | -4.68 | 1.38 | 2.92 | 27.15 |

均值：26.99

特效消耗：26.99-14.08=12.91ms

上限为50层时最适合，特效消耗：10.48ms

1. 华为Mate8 （中端机）：

0层：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DrawCall | CR(ms) | MSR(ms) | Culling(ms) | PU(ms) | PE(ms) | CPU(ms) |
| 150 | 18.13 | -2.17 | -2.2 | 0.05 | 0.02 | 13.83 |
| 149 | 18.62 | -2.83 | -1.8 | 0.05 | 0.02 | 14.06 |
| 155 | 18.9 | -3.77 | -1.4 | 0.04 | 0.01 | 13.78 |
| 154 | 18 | -2.26 | -1.67 | 0.04 | 0.01 | 14.12 |

均值：14.07

特效消耗：

50层：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DrawCall | CR(ms) | MSR(ms) | Culling(ms) | PU(ms) | PE(ms) | CPU(ms) |
| 224 | 25.33 | -1.95 | -3.45 | 0.66 | 1.13 | 21.72 |
| 227 | 24.48 | -2 | -3.14 | 0.53 | 0.76 | 20.63 |
| 230 | 28.33 | -2.05 | -4.09 | 0.62 | 0.98 | 23.79 |
| 230 | 27.61 | -2.23 | -3.2 | 0.59 | 0.67 | 23.44 |

均值：22.39

特效消耗：22.39-14.07=8.32ms

60层：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DrawCall | CR(ms) | MSR(ms) | Culling(ms) | PU(ms) | PE(ms) | CPU(ms) |
| 268 | 30.14 | -2.06 | -4.39 | 0.91 | 1.48 | 26.08 |
| 256 | 29.06 | -2.2 | -4.35 | 0.71 | 1.56 | 24.78 |
| 254 | 30.24 | -2.51 | -5.26 | 1.03 | 1.89 | 25.39 |
| 248 | 27.47 | -2.12 | -3.17 | 0.81 | 3.32 | 26.31 |

均值：25.46

特效消耗：25.46-14.07=11.39ms

70层：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DrawCall | CR(ms) | MSR(ms) | Culling(ms) | PU(ms) | PE(ms) | CPU(ms) |
| 272 | 33.34 | -2.05 | -4.64 | 1.08 | 4.43 | 32.16 |
| 291 | 30.48 | -2.14 | -4.32 | 0.84 | 1.08 | 25.94 |
| 293 | 37.51 | -2.13 | -7.82 | 0.99 | 3.14 | 31.69 |
| 295 | 30.93 | -2.05 | -3.99 | 1.13 | 1.43 | 27.45 |

均值：29.31

特效消耗：29.31-14.07=15.24ms

上限为60层时最适合，特效消耗：11.39ms

1. 三星S7（高端机）

0层：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DrawCall | CR(ms) | MSR(ms) | Culling(ms) | PU(ms) | PE(ms) | CPU(ms) |
| 174 | 12.85 | -1.66 | -0.66 |  |  | 10.53 |
| 173 | 13.71 | -1.56 | -0.98 | 0.02 | 0.01 | 11.2 |
| 174 | 13.08 | -1.52 | -0.68 |  |  | 10.88 |
| 174 | 14.73 | -1.53 | -1.12 |  |  | 12.08 |

均值：11.17

特效消耗：

50层：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DrawCall | CR(ms) | MSR(ms) | Culling(ms) | PU(ms) | PE(ms) | CPU(ms) |
| 265 | 22.08 | -1.65 | -2.8 | 0.61 | 2.59 | 20.83 |
| 265 | 20.99 | -1.57 | -3.43 | 0.5 | 0.3 | 16.79 |
| 261 | 21.54 | -1.54 | -3.59 | 0.47 | 0.47 | 17.35 |
| 277 | 21.05 | -1.81 | -2.7 | 0.61 | 1.11 | 18.26 |

均值：18.3

特效消耗：18.3-11.17=7.13ms

70层：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DrawCall | CR(ms) | MSR(ms) | Culling(ms) | PU(ms) | PE(ms) | CPU(ms) |
| 298 | 22.82 | -1.54 | -3.86 | 0.71 | 1.06 | 19.19 |
| 285 | 21.75 | -1.51 | -2.74 | 0.92 | 1.73 | 20.15 |
| 295 | 22.32 | -1.62 | -3.17 | 0.71 | 1.1 | 19.34 |
| 297 | 23.26 | -1.46 | -4.43 | 0.74 | 1.34 | 19.45 |

均值：19.53

特效消耗：19.53-11.17=8.36ms

80层：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DrawCall | CR(ms) | MSR(ms) | Culling(ms) | PU(ms) | PE(ms) | CPU(ms) |
| 312 | 24.74 | -1.56 | -3.37 | 0.65 | 1.25 | 21.71 |
| 309 | 22.65 | -1.6 | -3.7 | 0.95 | 2.11 | 20.41 |
| 311 | 24.65 | -2.97 | -3.29 | 0.76 | 1.01 | 20.16 |
| 294 | 25.87 | -1.49 | -5.91 | 0.98 | 1.51 | 20.96 |

均值：20.81

特效消耗：20.81-11.17=9.64ms

90层：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DrawCall | CR(ms) | MSR(ms) | Culling(ms) | PU(ms) | PE(ms) | CPU(ms) |
| 311 | 25.07 | -1.46 | -3.44 | 0.73 | 0.91 | 21.81 |
| 323 | 24.07 | -1.55 | -3.68 | 0.95 | 2.26 | 22.05 |
| 309 | 25.36 | -1.53 | -4.89 | 1.22 | 3.8 | 23.96 |
| 313 | 25.35 | -1.57 | -4.04 | 3.89 | 0.54 | 24.17 |

均值：22.97

特效消耗：22.97-11.17=11.8ms

80层 20.81-11.17=9.64ms

总结：

根据上面测试表格。低中高端机型的层数大小分别为：

低端：50层

中端：60层

高端：80层

遇到问题：

1. 测试特效分层数量时发现很多受击、buf类的特效以及个别英雄技能的特效没有做特效分层配置。
2. 美术需要一个工具，导出特效是否进行了分层配置。
3. 单个特效的规范，标准。（高、中、低配置）。粒子数量限制，shader在特效中的使用规范（如含有双面渲染的shader使用数量限制）
4. Overdraw对性能消耗也有很大影响，后期会做深度分析。

待做：

1. 自动化测试：分析特效在移动端上的性能消耗工具。

2018.05.02添加中高低端最大层数配置之后策划检查效果：

有些buf类的特效没有配置特效分层，所以这次主要看角色技能特效的播放是否达到需求效果。

测试用例：5个号跑治安巡逻

1、高端、中端机基本不影响特效的效果表现，低端机能明显看出部分特效不显示，主要特效都能显示出来，低端机偶尔能感觉明显卡顿（不播特效也会有卡顿感所以应该不是特效造成的卡顿）。

后续工作：

1. 程序：工具支持，导出已配置和未配置特效分层的特效清单。
2. 策划：根据该清单，安排特效分层制作的优先级。
3. 美术：根据策划给出的优先级，按顺序制作特效分层配置。

问题：

1、buf类特效在怪多的情况下对性能影响较大，建议优先制作该种类特效的优先级，便于测试出加入特效分层功能后的效果。