1.缓存变量, 如下代码中, 缓存\_ui会节约三次操作

var *\_ui* = *mEffect*.ui;

*x* = *\_ui*.x \* *zoomPercent*;

*y* = *\_ui*.y \* *zoomPercent*;

*width* = *\_ui*.w \* *zoomPercent*;

*height* = *\_ui*.h \* *zoomPercent*;

验证: t2+t3 约等于t1, t2+t3\*4=16s, 约等于t4

// 1. all in one 5321 ms

var *\_ui*;

for(var *i*=0; *i*<1000000;*i*++){

*x* = *mEffect*.ui.x \* *zoomPercent*;

}

// 2. ref only 1457 ms

*\_ui* = *mEffect*.ui;

for(*i*=0; *i*<1000000;*i*++){

*\_ui* = *mEffect*.ui;

}

// 3. set only 3656 ms

*\_ui* = *mEffect*.ui;

for(*i*=0; *i*<1000000;*i*++){

*x* = *\_ui*.x \* *zoomPercent*;

}

// 4. one for four 15097 ms

for(*i*=0; *i*<1000000;*i*++){

*\_ui* = *mEffect*.ui;

*x* = *\_ui*.x \* *zoomPercent*;

*y* = *\_ui*.y \* *zoomPercent*;

*width* = *\_ui*.w \* *zoomPercent*;

*height* = *\_ui*.h \* *zoomPercent*;

}

2.引用匿名函数还是使用有名函数, 数据没有显著差异, 无法解释为什么两者开头几次都耗时较多

for(var *i*=0; *i*< 10000000; *i*++){

if (this.mOnAudioEffectUpdatedListener !== null)

this.mOnAudioEffectUpdatedListener();

}

*mEffect*.setOnAudioEffectUpdatedListener(function(){});

qml: 25463 passed

qml: 26737 passed

qml: 24776 passed

qml: 18298 passed

qml: 17375 passed

qml: 16964 passed

qml: 18176 passed

function *func1*(){}

*mEffect*.setOnAudioEffectUpdatedListener(*func1*);

qml: 24765 passed

qml: 26349 passed

qml: 24372 passed

qml: 16897 passed

qml: 18022 passed

qml: 17448 passed

qml: 17509 passed

qml: 18395 passed

qml: 17481 passed

3.多一个函数操作效率低, 封装性好

// 72822 ms

*Item*.prototype.getSetting = function() {

return this.mConfig.setting;

}

for(var i=0; i< 1000000; i++)

{

bypassed = !mEffect.getSetting().enable;

}

// 56554 ms

for(var *i*=0; *i*< 10000000; *i*++)

{

*bypassed* = !*mEffect*.mConfig.setting.enable;

}