Figure - Bug #115534

测试 Test-ST # 113421 (New): V4.0功能与专项测试

测试 Test-ST # 113423 (New): V4.0专项--BSP专项--稳定性-各项功能长时间运行

【AF】【EVT】【稳定性】进行MTBF脚本长时间运行，导致屏幕无法响应

2023-01-30 11:48 - 物联网测试组_CDTS 段小刚

Status:	CLOSED	**Start date:**	2023-01-30
Priority:	High	**Due date:**	2023-03-15
Assignee:	CDTS_Test 吴诗雨	**% Done:**	100%
Category:	CD-FW	**Estimated time:**	0.00 hour
Target version:	VC1_FSE_0094_20230425		
Need_Info:	TEST	**Found Version:**	FlatBuild_HH_VX1_MCE_FSE.M.D.user.01.00.X101.2
Resolution:	FIXED	**Degrated:**	No
Severity:	Critical	**Verified Version:**	
Reproducibility:	Occasionally	**Fixed Version:**	2023-04-26
Test Type:	IT	**Root cause:**	原生就存在的layer过多问题，添加了改善机制，定

Description

【前提条件】
1、设备成功启动
2、确保MTBF脚本运行的前置条件已准备

【测试步骤】
1、使用python运行对应脚本
2、运行7*24小时
3、查看设备运行结果

【预期结果】
3、设备正常运行，没有崩溃，重启等现象
【实际结果】
3、屏幕点击后，系统无法做出响应

【复现率】
1/2

History

#1 - 2023-01-30 16:05 - 物联网测试组_CDTS 段小刚

- File tlog_iov020101750000616202541 1a2212290000000841_213_0130113410.tar.gz added

#2 - 2023-01-30 17:35 - CD TPM-王祥林

- Category changed from BSP to CD-APP

- Assignee changed from CD TPM-王祥林 to CD APP-王营

王营

根据测试反馈，测试机停留在屏保界面，点击无反应。通过getevent命令查看底层touch事件是有上报的，通过adb命令发送power key息屏后单击屏幕也可以亮屏。
看起来像是屏保界面出了异常，帮忙调查一下。

#3 - 2023-01-30 17:42 - 物联网测试组_CDTS 段小刚

- File 202301201023.log added

#4 - 2023-01-31 09:52 - CD APP-王营

- Due date set to 2023-02-09

- Status changed from New to ASSIGNED

#5 - 2023-02-09 19:46 - CD APP-王营

- Due date changed from 2023-02-09 to 2023-02-15

■ Current conclusion
问题分析中
■ My analysis
通过当前log分析，问题可能出在native层，MotionEvent在传递时数据有异常。

■ Next action
继续分析log，持续跟进

#6 - 2023-02-16 09:30 - CD APP-王营

- Due date changed from 2023-02-15 to 2023-02-17

#7 - 2023-02-17 09:59 - CD TPM-王祥林

- Target version set to VC1_FSE_0078_20230228

#8 - 2023-02-20 09:14 - CD APP-王营

- Due date changed from 2023-02-17 to 2023-02-24

#9 - 2023-02-20 20:29 - CD APP-王营

- Assignee changed from CD APP-王营 to CD FW王武军

#10 - 2023-02-22 10:04 - CDTS_Test 吴诗雨

2月21日跑mtbf 16小时，出现卡在屏保的情况。可以下划打开状态栏，通知中心，但是没办法通过点击或者长按进入设置。tlog已经发给研发。

#11 - 2023-02-24 16:36 - CD FW王武军

【】当前状态

根据测试提供的日志和现象截图，确认了问题

【】分析情况

1、测试反馈的现象是发生问题时，屏幕无响应。我怀疑可能存在ANR，查看了日志，并未找到有关的ANR信息，所以问题应该不是由ANR引起。

2、我怀疑是进程发生了奔溃，分析了tombstone文件，从tombstone中发现了类似这样的日志信息：
Build fingerprint: 'TC/figure_CN/figure:12/SKQ1.220201.001/2134:user/test-keys'
Revision: '0'
ABI: 'arm64'
Timestamp: 2023-02-21 07:52:11.627199597+0800
Process uptime: 0s
Cmdline: /system/bin/audioserver

pid: 898, tid: 1627, name: TimeCheckThread >>> /system/bin/audioserver <<<
uid: 1041
signal 6 (SIGABRT), code ~~1 (SI_QUEUE), fault addr -------~~
Abort message: 'TimeCheck timeout for IAudioFlinger command 38'
x0 0000000000000000 x1 000000000000065b x2 0000000000000006 x3 0000006e1e3d0530
x4 6d686b456e686374 x5 6d686b456e686374 x6 6d686b456e686374 x7 7f7f7f7f7f7f7f7f
x8 00000000000000f0 x9 117ed7971eb9a4ed x10 0000000000000000 x11 ffffff80ffffffdf
x12 0000000000000001 x13 000000000000002f x14 0000006e1e3d06c0 x15 0000000034155555
x16 0000007142246060 x17 0000007142222560 x18 0000006e1e1d0000 x19 0000000000000382
x20 000000000000065b x21 00000000ffffffff x22 b400006e5fbea010 x23 b400006e3fbe3c54
x24 b400006e3fbe3c54 x25 0000006e1e3d0cb0 x26 0000006e1e3d0ff8 x27 00000000000fc000
x28 0000006e1e2d8000 x29 0000006e1e3d05b0
lr 00000071421d295c sp 0000006e1e3d0510 pc 00000071421d2988 pst 0000000000001000

进行了堆栈分析，发现出现问题的代码位置是：
http://192.168.87.66:8006/source/xref/Pre_figure_turbox-c2130c-la1.1-qssi12-dev/LA.QSSI/LINUX/android/bionic/libc/bionic/abort.cpp#50
但是从代码来看与业务逻辑没有太大关系，从这个角度也没有发现有效的问题点。

3、从具体的日志文件出发寻找可疑的日志，发现有如下的日志

02-21 08:04:40.375 1521 1632 E SurfaceComposerClient: SurfaceComposerClient::createSurface error Out of memory
02-21 08:04:40.375 1521 1632 E WindowManager: Unhandled exception while laying out windows
02-21 08:04:40.375 1521 1632 E WindowManager: android.view.Surface$OutOfResourcesException
02-21 08:04:40.375 1521 1632 E WindowManager: at android.view.SurfaceControl.nativeCreate(Native Method)
02-21 08:04:40.375 1521 1632 E WindowManager: at android.view.SurfaceControl.<init>(SurfaceControl.java:1435)
02-21 08:04:40.375 1521 1632 E WindowManager: at android.view.SurfaceControl.<init>(SurfaceControl.java:87)
02-21 08:04:40.375 1521 1632 E WindowManager: at android.view.SurfaceControl$Builder.build(SurfaceControl.java:1108)
02-21 08:04:40.375 1521 1632 E WindowManager: at com.android.server.wm.SurfaceAnimator.createAnimationLeash(SurfaceAnimator.java:424)
02-21 08:04:40.375 1521 1632 E WindowManager: at com.android.server.wm.SurfaceAnimator.startAnimation(SurfaceAnimator.java:173)
02-21 08:04:40.375 1521 1632 E WindowManager: at com.android.server.wm.WindowContainer.startAnimation(WindowContainer.java:2526)
02-21 08:04:40.375 1521 1632 E WindowManager: at com.android.server.wm.WindowContainer.applyAnimationUnchecked(WindowContainer.java:2757)
02-21 08:04:40.375 1521 1632 E WindowManager: at com.android.server.wm.Task.applyAnimationUnchecked(Task.java:4062)
02-21 08:04:40.375 1521 1632 E WindowManager: at com.android.server.wm.WindowContainer.applyAnimation(WindowContainer.java:2628)
02-21 08:04:40.375 1521 1632 E WindowManager: at com.android.server.wm.AppTransitionController.applyAnimations(AppTransitionController.java:591)
02-21 08:04:40.375 1521 1632 E WindowManager: at com.android.server.wm.AppTransitionController.applyAnimations(AppTransitionController.java:723)
02-21 08:04:40.375 1521 1632 E WindowManager: at com.android.server.wm.AppTransitionController.handleAppTransitionReady(AppTransitionController.java:224)
02-21 08:04:40.375 1521 1632 E WindowManager: at com.android.server.wm.RootWindowContainer.checkAppTransitionReady(RootWindowContainer.java:1030)
02-21 08:04:40.375 1521 1632 E WindowManager: at
com.android.server.wm.RootWindowContainer.performSurfacePlacementNoTrace(RootWindowContainer.java:869)
02-21 08:04:40.375 1521 1632 E WindowManager: at com.android.server.wm.RootWindowContainer.performSurfacePlacement(RootWindowContainer.java:805)
02-21 08:04:40.375 1521 1632 E WindowManager: at com.android.server.wm.WindowSurfacePlacer.performSurfacePlacementLoop(WindowSurfacePlacer.java:177)
02-21 08:04:40.375 1521 1632 E WindowManager: at com.android.server.wm.WindowSurfacePlacer.performSurfacePlacement(WindowSurfacePlacer.java:126)
02-21 08:04:40.375 1521 1632 E WindowManager: at com.android.server.wm.WindowSurfacePlacer.performSurfacePlacement(WindowSurfacePlacer.java:115)
02-21 08:04:40.375 1521 1632 E WindowManager: at com.android.server.wm.DisplayContent.layoutAndAssignWindowLayersIfNeeded(DisplayContent.java:3585)
02-21 08:04:40.375 1521 1632 E WindowManager: at com.android.server.wm.WindowManagerService$H.handleMessage(WindowManagerService.java:5335)
02-21 08:04:40.375 1521 1632 E WindowManager: at android.os.Handler.dispatchMessage(Handler.java:106)
02-21 08:04:40.375 1521 1632 E WindowManager: at android.os.Looper.loopOnce(Looper.java:201)
02-21 08:04:40.375 1521 1632 E WindowManager: at android.os.Looper.loop(Looper.java:288)
02-21 08:04:40.375 1521 1632 E WindowManager: at android.os.HandlerThread.run(HandlerThread.java:67)
02-21 08:04:40.375 1521 1632 E WindowManager: at com.android.server.ServiceThread.run(ServiceThread.java:44)

从日志的信息来看是有surfaceflinger相关的日志错误，计划从这个角度进行一下追踪。

【】计划策略

1、目前在跟进monkey相关的bug，该问题暂时未继续跟进。

2、计划从surfaceflinger的错误日志进行一下追踪，看是否能找到一些突破点。

#12 - 2023-02-27 14:45 - CD APP-王营

- Due date changed from 2023-02-24 to 2023-03-15

#13 - 2023-02-28 14:22 - CDTS-TEST 周婷

- Severity changed from Normal to Major

#14 - 2023-02-28 14:23 - CD TPM-王祥林

- Target version changed from VC1_FSE_0078_20230228 to VC1_FSE_0082_20230314

#15 - 2023-03-02 11:44 - CD BSP-杜磊

- Subject changed from 【BSP】【EVT】【稳定性】进行MTBF脚本长时间运行，导致屏幕无法响应 to 【AF】【EVT】【稳定性】进行MTBF脚本长时间运行，导致屏幕无法响应

#16 - 2023-03-10 18:32 - CD FW王武军

【】当前状态

从SurfaceComposerClient: SurfaceComposerClient::createSurface error Out of memory展开逻辑分析

触发日志点：

```
status_t SurfaceComposerClient::createSurfaceChecked(){
  err = mClient->createSurface(name, w, h, format, flags, parentHandle, std::move(metadata),
  &handle, &gbp, &id, &transformHint);
  if (outTransformHint) {
   *outTransformHint = transformHint;
  }
  ALOGE_IF(err, "SurfaceComposerClient::createSurface error %s", strerror(-err));
 }
status_t SurfaceFlinger::addClientLayer(){
 if (mNumLayers >= ISurfaceComposer::MAX_LAYERS) {
  ALOGE("AddClientLayer failed, mNumLayers (zu) >= MAX_LAYERS (%zu)", mNumLayers.load(),
        ISurfaceComposer::MAX_LAYERS);
    mCurrentState.traverseInZOrder([x%x](Layer* layer) {
        const auto& p = layer->getParent();
        ALOGE("layer (%s) ::  parent (%s).",
        layer->getName().c_str(),
        (p != nullptr) ? p->getName().c_str() : "no-parent");
    });
    return NO_MEMORY;
  }
 }
```

mNumLayers值的异常引发问题，而该值的调整逻辑与下面的方法有关系：

void SurfaceFlinger::onLayerFirstRef(Layer* layer) {}

void SurfaceFlinger::onLayerDestroyed(Layer* layer) {}

发生问题时：

通过脚本配合日志分析发现，在启动应用或者切换窗口时，添加的layer数与移除的layer数

并不是成正相关，随着脚本测试时间的持续，已经存在的layer数会越来越大最后超出阈值，报出异常信息。

【】后续策略
在正常启动应用和切花应用时，layer的数不会持续无限的增加，所以怀疑该问题的发生与测试所执行
测试脚本有强相关性，后续计划从系统逻辑层面结合测试的脚本细节展开，分析每一个测试点在逻辑层面的具体表现，
看能否定位出是什么具体的测试行为会触发问题或者引起问题的发生。

#17 - 2023-03-15 14:25 - CD FW王武军

【】当前状态
根据之前的怀疑点：layer的慢慢增加最后超出了限制，与测试脚本的逻辑有强关系
一、我分析了测试的脚本源码：
1）我发现存在类似：time.sleep(1)、self.device.implicitly_wait(3)
从代码来看，应该是脚本在等在上一次事件或者操作的响应和执行，但是这个事件从设备的
层面也许时间可能并不能如预期，有时候逻辑层面的等待，并不是系统正常执行消耗的具体时间，所以在主动等待后
执行后续的逻辑也许并不能达到预期的效果。
疑问点：这个时间是否能达到预期，特别是长时间测试时？
2）是否完成了模块的单元测试
比如：小窗测试、游戏测试、音乐播放、视屏播放，是否可以做一个单元测试，单独的把每一个模块单独的跑一下脚本？

二、为了追踪问题，我在能反馈出layer数量变化的地方添加了日志信息，来监layer数量的变化
void SurfaceFlinger::onLayerFirstRef(Layer* layer) {}
void SurfaceFlinger::onLayerDestroyed(Layer* layer) {}
分析日志提交：
https://dev.thundercomm.com/gerrit/c/general/platform/frameworks/base/+/187456
https://dev.thundercomm.com/gerrit/c/general/platform/frameworks/native/+/187455
编译的Jenki版本：
//userdebug 版本
/Pre_figure/VerifyBuild/Pre_figure_turbox-c2130c-la1.1-qssi12-dev/20230314/202303141719-2928
//user版本
/Pre_figure/VerifyBuild/Pre_figure_turbox-c2130c-la1.1-qssi12-dev/20230314/202303141750-2929

请测试同事使用Jenkins的userdebug版本，帮忙做如下针对性测试：
1）不跑脚本，用手动的方式，执行脚本中的测试流程，测试轮数最好在10轮以上，记录日志；
2）将测试脚本进行拆分，最好能做单元的测试，如果不能完全的单元，可以把几个功能单独放在一起分别进行测试，
几个重要的模块：小窗相关的测试、音视频相关的测试、游戏网络相关的测试，记录日志；
3）直接用脚本跑测试，记录日志；
（Tisp：1、跑脚本测试时，执行时间不用很长，可以控制在30分钟到60分钟；2、记录下做每项测试的开始时间和结束时间）

#18 - 2023-03-20 17:39 - CDTS_Test 吴诗雨

已根据要求做针对性测试，log已发送

#19 - 2023-03-23 14:09 - CDTS_TEST 王成

- Target version changed from VC1_FSE_0082_20230314 to VC1_FSE_0090_20230411

#20 - 2023-03-23 15:15 - CD FW王武军

- Target version changed from VC1_FSE_0090_20230411 to VC1_FSE_0082_20230314

【】当前状态

1、分析了mtbf-log的测试日志，对比了单元测试信息和手动测试信息，得出如下信息

1）小窗测试模块，随着测试的不停进行，layer存在异常增加；
2）脚本中如果通过类似click的方式执行某个view的点击操作，如果view不能正确响应，也会导致layer的异常增加，如：

```
Traceback (most recent call last):
  File "D:\Figure\Fmtbf1\Fmtbf\TestScript.py", line 649, in run
    self.Music_play_test()
  File "D:\Figure\Fmtbf1\Fmtbf\TestScript.py", line 241, in Music_play_test
    self.device.xpath('//*[@resource-id="com.android.music:id/play_indicator"]').click()
  File "D:\python\lib\site-packages\uiautomator2\xpath.py", line 595, in click
    el = self.get(timeout=timeout)
  File "D:\python\lib\site-packages\uiautomator2\xpath.py", line 524, in get
    raise XPathElementNotFoundError(self._xpath_list)
uiautomator2.exceptions.XPathElementNotFoundError: ['//*[@resource-id="com.android.music:id/play_indicator"]']
```

2、后续需要从两个方面来跟进该问题

1）研发 跟进小窗测试模块中所涉及的功能，分析为什么在这行小窗相关的操作时，layer会异常增加；
2）测试 需要确认脚本中，所涉及到的类似click这样的操作时，如果脚本未正常反馈时，对比手动测试正常点击，看是否存在功能流程的问题，
如果相同的操作手动点击没有问题，那么是否需要调整脚本或者这个测试项不需要用脚本来验证，需要测试同事来考虑下。

#21 - 2023-03-31 10:54 - CD APP-王营

- Category changed from CD-APP to CD-FW

#22 - 2023-04-03 13:57 - CD FW王武军

【】当前状态

小窗相关测试引起layer异常增加的调查

1）对比测试分析
a、不考虑高合特殊定制逻辑的情况下，针对纯粹的小窗功能做测试，layer同样异常增加；
b、使用同样搭载原生代码的其他设备进行相同的测试，问题表现一样，layer异常增加;
c、使用12版本的pixel进行同样的测试，未出现layer异常增加的表现
初步结论：在原生代码中开启小窗，依附原生的代码实现时，针对小窗的测试，都会存在layer异常增加；
而pixel虽然都是12的代码，但是pixel应该是进行了优化，所以未出现问题。

2）后续策略

a、先解决小窗基本功能不出现layer的异常增加，再检查包含高合定制功能是否会有layer异常增加；

b、解决小窗基本功能中出现的layer异常增加的两个方向：上层是否合理构建了正确的layer、底层是否正常的销毁了应该被销毁的layer

#23 - 2023-04-06 11:16 - CDTS_Test 吴诗雨

当前进度情况如下：
进行了首轮删除小窗模块后的mtbf跑测，正常运行的时间相对延长，三台设备，两台设备运行了39小时（仍正常运行），一台设备运行了42小时。
运行时间较长的设备（42小时）出现了黑屏卡住的情况，经分析是由于系统异常报错，造成layer过多。

目前mtbf出现的问题黑屏卡住等情况是layer过多造成的，经讨论，产生此情况的原因以及相对应的处理如下：
1、系统长时间进行mtbf测试时，响应时间会变长，此时脚本下发点击动作，会造成异常报错，从而产生过多layer。该报错没有固定出现的时间和地点，

故需要考虑是否在脚本添加点击之前的判断或者延长的等待时间
2、小窗本身的逻辑问题，造成layer过多的情况。脚本将暂时去除掉小窗部分进行跑测，等待研发进行修改后再加入。

#24 - 2023-04-06 14:24 - CDTS_Test 吴诗雨

三台设备中，另外两台设备跑测到现在共44小时，未出现明显问题。

#25 - 2023-04-06 17:31 - CDTS_TEST 王成

- Target version changed from VC1_FSE_0082_20230314 to VC1_FSE_0094_20230425

#26 - 2023-04-10 14:29 - CDTS_TEST 王成

- Priority changed from Normal to High
- Severity changed from Major to Critical

#27 - 2023-04-10 18:03 - CD FW王武军

【】当前状态
唐军的调查进度：
1, 从问题单的log看, 看起来是Task没有销毁, 可能是system没有丢弃对BBinder的引用.
2, 从自测20次看, Offscreen Layers存在很多重复的Task Surface没有销毁. 看起来这是当前问题的另一种表现形式, Task的销毁没有完成.

下一步:
1, 确认"可能是system没有丢弃对BBinder的引用"和Task销毁逻辑.
2, 请测试帮忙在快要出现问题时, 抓取adb shell dumpsys activity activities > ams, adb shell dumpsys window windows > wms和adb shell dumpsys SurfaceFlinger > sf;

#28 - 2023-04-11 18:41 - CD FW王武军

【】当前状态

唐军的调查进展：

1, 多次自测(20次),未释放的layer与Offscreen Layers基本一致.
主要是下面两种Layer未释放. Task数量基本是WallpaperWindowToken的两倍.
Surface(name=Task=19)/@0x3d1aa49 - animation-leash of recents_animation#0
Surface(name=WallpaperWindowToken{55f4d5e token=android.os.Binder@4047699})/@0x947c7ea - animation-leash of window_animation#0

2, 脚本跑15个小时左右时, Visible layers (count = 2603). 与自测20次的规律一致.

3, Google在12L上有发现max layer count问题,并且可能T也有,为了定位问题, 在T上有加入更多该问题的dump信息.
commit 26b98f4271095999a721fd85f17044eb8441cc97
Author: Robert Carr <racarr@google.com>
Date: Mon Apr 11 15:54:31 2022 -0700
    SurfaceFlinger: Dump on layer leaks
    Every so often some process will leak layer references and
      SurfaceFlinger will eventually crash (or refusing to create
      new layers because of the max layer count). We never have
      enough info to trace these down, and it becomes a horrible
      game of chasing repro steps. While we currently have no
      complaints about anything like this in T, a recent
      12L bug serves as a timely reminder that we should get

this logging in BEFORE the next issue occurs, rather
than after.
Bug: 227286456
Test: Existing tests pass
Change-Id: I2c8b834c93a27204225b482fea1e54837204c9cf

4, "animation-leash of recents_animation"与其它"animation-leash"的reset有差异.
5, 打入Android T的Leash相关patch验证无效.

下一步: 继续上面第4点以及其它尝试.

#29 - 2023-04-12 18:48 - CD FW王武军

【】当前状态

唐军的调查进展：

1, 对比了T和S的surfaceflinger相关逻辑, 有6个patch待验证. 要么surfaceflinger崩溃, 要么验证没效果.
2, 在OnAnimationFinishedCallback中reset(false)时, Offscreen Layer不再增加, 但是mNumLayers仍然没减少,即Layer未释放.
Task的leash先挂到DefaultTaskDisplayArea, 接着Task从DefaultTaskDisplayArea之下移动到leash之下.
在RecentsAnimation.finishAnimation时,Task从leash之下移动到DefaultTaskDisplayArea之下.

注:在OnAnimationFinishedCallback中reset(true)时, Offscreen Layer会增加两种Layer.
Task的leash先挂到DefaultTaskDisplayArea, 接着Task从DefaultTaskDisplayArea之下移动到leash之下.
在RecentsAnimation.finishAnimation时,Task从leash之下移动到DefaultTaskDisplayArea之下.leash从DefaultTaskDisplayArea之下移除.

3, 如果对下面两种Layer不做动画, 可以规避问题. 但是效果不清楚.
Surface(name=Task=19)/@0x3d1aa49 - animation-leash of recents_animation#0
Surface(name=WallpaperWindowToken{55f4d5e token=android.os.Binder@4047699})/@0x947c7ea - animation-leash of window_animation#0

4, wms逻辑看着是正常的, 因为有销毁动作. 而该问题根源是Layer没有被销毁. 强制对Offscreen Layer的特定Layer销毁mNumLayers会减少.
看起来是系统进程一直持有Handle引用未释放导致了问题的发生.

下一步:
研究正常的leash销毁流程来找到规避方法或者找到差异点.

#30 - 2023-04-13 18:41 - CD FW王武军

【】当前状态

唐军的调查进展：

1, 追踪surfaceflinger中的Layer存储和销毁流程.
Task Layer在重新挂载到DefaultDisplay时在leash中remove失败.
Layer有5种销毁流程. 与我们相关的销毁有两种, 我们的销毁缺少对BBinder的解引用.
正在确认正常销毁的相似的Layer流程, 然后根据wms信息或者Offscreen Layers信息给出规避方法.

#31 - 2023-04-14 18:58 - CD FW王武军

【】当前状态

唐军的调查进展：

看起来是binder调用出了问题, surfaceflinger的Layer销毁流程看起来正确。
正在做规避patch。

#32 - 2023-04-15 13:38 - CD FW 曹覃刚

- Status changed from ASSIGNED to NEED_INFO
- Assignee changed from CD FW王武军 to CDTS_Test 吴诗雨

Hi 诗雨

> *2、小窗本身的逻辑问题，造成layer过多的情况。脚本将暂时去除掉小窗部分进行跑测，等待研发进行修改后再加入。*

针对问题发生的原因二，我们做了修复，并编译了验证版本
/Pre_figure/VerifyBuild/Pre_figure_turbox-c2130c-la1.1-qssi12-dev/20230415/202304151042-3294
请帮忙使用此版本验证MTBF脚本长时间运行，是否还有Layer过多的黑屏问题

#33 - 2023-04-18 10:04 - CDTS_Test 吴诗雨

- File tlog_273f7c37_000233_0418094114.tar.gz added

验证情况：
未出现黑屏，出现多次重启的问题。

#34 - 2023-04-18 16:17 - CD FW 曹覃刚

Hi 诗雨
已优化patch内容，并重新编译了版本，请使用新版本验证，谢谢
/Pre_figure/VerifyBuild/Pre_figure_turbox-c2130c-la1.1-qssi12-dev/20230418/202304181109-3312

#35 - 2023-04-19 10:20 - CDTS_Test 吴诗雨

VB验证情况：
一台设备跑测一晚上15小时
出现一次黑屏，出现多次重启。

#36 - 2023-04-19 19:04 - CDTS_Test 吴诗雨

- Assignee changed from CDTS_Test 吴诗雨 to CD FW 曹覃刚

#37 - 2023-04-19 19:51 - CD FW 曹覃刚

- Assignee changed from CD FW 曹覃刚 to CDTS_Test 吴诗雨

Hi 诗雨

> *VB验证情况：*

> *一台设备跑测一晚上15小时*
> *出现一次黑屏，出现多次重启*

上次编译的版本时间有误，重新编译了版本，目录如下，该版本正在同步中，请使用该版本再次测试MTBF脚本，十分感谢
VerifyBuild/Pre_figure_turbox-c2130c-la1.1-qssi12-dev/20230419/202304191258-3333

#38 - 2023-04-20 19:55 - CDTS_Test 吴诗雨

昨晚跑测DB版本带小窗，今晚用VB进行跑测

#39 - 2023-04-21 10:08 - CDTS_Test 吴诗雨

验证情况：
一台设备跑测了14小时
看截图出现了数次黑屏的情况，出现在由小窗切换成mini小窗的时候，未出现重启的情况。
日志比较大，已经私发

#40 - 2023-04-21 16:39 - CDTS_Test 吴诗雨

- Assignee changed from CDTS_Test 吴诗雨 to CD FW 曹覃刚

#41 - 2023-04-21 17:45 - CD FW 曹覃刚

■ 当前的状态

> *看截图出现了数次黑屏的情况，出现在由小窗切换成mini小窗的时候，未出现重启的情况。*

已定位问题原因，正在优化patch内容，预计今晚会再编译一个版本，应该是最终版本了

■ 下一步计划
版本编译完成后，跑MTBF验证

#42 - 2023-04-23 09:40 - CD FW 曹覃刚

- Assignee changed from CD FW 曹覃刚 to CDTS_Test 吴诗雨

Hi 诗雨
新版本，已编译完成，请帮助跑MTBF验证，谢谢
/Pre_figure/VerifyBuild/Pre_figure_turbox-c2130c-la1.1-qssi12-dev/20230421/202304212019-3399

#43 - 2023-04-24 14:55 - CDTS_Test 吴诗雨

验证情况：
跑测14小时，出现一次重启，未出现黑屏卡住的情况，再跑测24小时查看情况

#44 - 2023-04-25 11:04 - CD TEST-方永红

- Assignee changed from CDTS_Test 吴诗雨 to CD FW 曹覃刚

VB验证情况：
跑测24小时，出现数次重启，重启原因是由于小窗频繁引动切换引起的重启，已有bug跟踪。
未出现黑屏卡住的情况，验证成功

#45 - 2023-04-25 19:21 - IoT scm

Gerrit Merge Information：

ID	Project	Branch	Uploader
194038	general/platform/frameworks/native	Pre_figure_turbox-c2130c-la1.1-qssi12-dev	caoqg0702@thundersoft.com
FW:SystemOptimization: Improvement of abnormal increase in LayerTC-RID: 1201-0205101IssueID: TS-R-BUG-115534Change-Id: Idff5066985cf7ae6afaffb90abc55a62ea0da26a			

#46 - 2023-04-25 19:23 - CD FW 曹覃刚

- Status changed from NEED_INFO to RESOLVED
- Assignee changed from CD FW 曹覃刚 to CDTS_Test 吴诗雨
- % Done changed from 0 to 100
- Resolution changed from -- to FIXED
- Degrated changed from -- to No
- Fixed Version set to 2023-04-26
- Root cause set to 原生就存在的layer过多问题，添加了改善机制，定时清除异常的layer

#47 - 2023-04-26 16:04 - CDTS_Test 吴诗雨

刷4.26DB进行新一轮的MTBF跑测中

#48 - 2023-05-05 10:49 - CDTS_Test 吴诗雨

- Status changed from RESOLVED to VERIFIED

MTBF长时间跑测，暂未出现屏幕无法响应的情况。

#49 - 2023-05-05 10:49 - CDTS_Test 吴诗雨

- Status changed from VERIFIED to CLOSED

Files

tlog_iov020101750000616202541 1a2212290000000841_213_0130113410.tar.gz	19 MB	2023-01-30	物联网测试组_CDTS 段小刚
202301201023.log	14.9 MB	2023-01-30	物联网测试组_CDTS 段小刚
tlog_273f7c37_000233_0418094114.tar.gz	38.2 MB	2023-04-18	CDTS_Test 吴诗雨