Screen flickering issue while attaching SurfaceView

Fixed Bug P4 + Add Hotlist

STATUS UPDATE No update yet. Edit

DESCRIPTION [Deleted User] created issue #1 Apr 23, 2020 08:43PM

CAMERAX VERSION:
def camera_x_version = '1.0.0-beta03'
def camera_x_view_ver = '1.0.0-alpha10'

implementation "androidx.camera:camera-core:$camera_x_version"
// CameraX-Camera2 extensions
implementation "androidx.camera:camera-camera2:$camera_x_version"
// CameraX Lifecycle
implementation "androidx.camera:camera-lifecycle:$camera_x_version"
// CameraX View
implementation "androidx.camera:camera-view:$camera_x_view_ver"

CAMERA APPLICATION NAME AND VERSION:
N.A

ANDROID OS BUILD NUMBER:
ANY

DEVICE NAME:
ANY

DESCRIPTION:
While we're trying to open the camera preview, there is a short flickering which comes every second time and onwards we randomly try to close and open the camera activity.

We've tested it with Implementation mode as PreviewView.ImplementationMode.TEXTURE_VIEW and it works fine.

LIST ANY EXPERIMENTAL FEATURES:
N.A

STEPS TO REPRODUCE:
1. Add the camera activity on a button click in say Activity A.
2. Open the camera the first time.
3. Back press and close immediately.
4. Click on the button again.
5. Repeat steps 3 to 4 very quickly.

OBSERVED RESULTS:
While the activity is opened, there is a solid bg color seen that was applied, and just at the time when SurfaceView is created and attached with the preview, the screen looks transparent for a very short while and the Activity A is seen on the background. After that, the preview appears and keeps running.

EXPECTED RESULTS:
There should be no transparent flickering as such. TextureView implementation works well in this case.

REPRODUCIBILITY: (5 of 5, 1 of 100, etc)
5 out of 10.

ADDITIONAL INFORMATION:
Seems that some resource is not being closed properly which is why this issue is coming on the very first time.

CODE FRAGMENTS (this will help us troubleshoot your issues):

```
cameraProvider.unbindAll()

try {
   val camera = cameraProvider.bindToLifecycle(this, cameraSelector, preview, imageCapture)
   preview?.setSurfaceProvider(viewFinder?.createSurfaceProvider(camera.cameraInfo))
} catch (exc: Exception) {
   Log.e(TAG, "Some exception in the use cases : $exc")
}
```

SOME LOGCAT:

```
E/CameraService: checkTemperature (0)
E/CameraService: setCompanionDisableSysFs : cc.disable property set to 0
E/ExynosCamera3: Build Date is (Mar  3 2020) (19:42:55)
E/CameraService: onTorchStatusChangedLocked: cannot get torch status of camera 1: No such file or directory (-2)
E/CameraService: onTorchStatusChangedLocked: cannot get torch status of camera 1: No such file or directory (-2)
```

Reporter	[Deleted User]
Type	Bug
Priority	P4
Severity	S4
Status	Fixed
Access	Default access View
Expanded Access	
Assignee	hu...@google.com
Verifier	--
Collaborators	
CC	[Deleted User] hu...@google.com xi...@google.com
AOSP ID	--
Estimate	--
InReview	--
Pending	--
Targeted To	--
TF Estimate	--
Found In	--
Targeted To	may-2020
Verified In	--
In Prod	

Show 1 additional field

```
E/ExynosCamera3Interface: INFO(HAL3_camera_device_initialize[250]): dual cam_state[0](3)
E//vendor/bin/hw/vendor.samsung.hardware.camera.provider@3.0-service: Failed to get IAshmemDeviceService.
E//vendor/bin/hw/vendor.samsung.hardware.camera.provider@3.0-service: Failed to get IAshmemDeviceService.
E/Camera2-FrameProcessorBase: FrameProcessorBase: created
E/Camera2-FrameProcessorBase: registerListener: Registering listener for frame id range 0 - 2147483647
E/ExynosCameraRequestManager: [CAM_ID(1)][]-ERR(m_deleteStreamThread[3961]):failed, StreamThreadMap size is ZERO, size(0)
E/ExynosCameraBufferManager: [CAM_ID(1)][ISP_RE_BUF]-ERR(m_defaultAlloc[1486]):buffer[0].fd[1] = 14 already allocated
E/ExynosCameraBufferManager: [CAM_ID(1)][ISP_RE_BUF]-ERR(m_defaultAlloc[1486]):buffer[0].fd[0] = 15 already allocated
E/ExynosCameraBufferManager: [CAM_ID(1)][YUV_CAPTURE_BUF]-ERR(m_defaultAlloc[1486]):buffer[0].fd[1] = 16 already allocated
E/ExynosCameraBufferManager: [CAM_ID(1)][YUV_CAPTURE_BUF]-ERR(m_defaultAlloc[1486]):buffer[0].fd[0] = 17 already allocated
E/ExynosCameraBufferManager: [CAM_ID(1)][THUMBNAIL_BUF]-ERR(m_defaultAlloc[1486]):buffer[0].fd[1] = 18 already allocated
E/ExynosCameraBufferManager: [CAM_ID(1)][THUMBNAIL_BUF]-ERR(m_defaultAlloc[1486]):buffer[0].fd[0] = 19 already allocated
E/BufferQueueProducer: [com.xxxx.xxxxx.debug/com.xxxx.xxxxx.auth.activities.LoginActivity$_16221#0] disconnect: not connected
(req=1)
E/WindowManager: win=Window{f89099 u0 com.xxxx.xxxxx.debug/com.xxxx.xxxxx.auth.activities.LoginActivity} destroySurfaces:
appStopped=true win.mWindowRemovalAllowed=false win.mRemoveOnExit=false win.mViewVisibility=8
caller=com.android.server.wm.AppWindowToken.destroySurfaces:1189
com.android.server.wm.AppWindowToken.destroySurfaces:1170 com.android.server.wm.AppWindowToken.notifyAppStopped:1225
com.android.server.wm.ActivityRecord.activityStoppedLocked:2607
com.android.server.wm.ActivityTaskManagerService.activityStopped:2356 android.app.IActivityTaskManager$Stub.onTransact:2183
android.os.Binder.execTransactInternal:1021
E/SemCamera-JNI-Java: SemCamera.open()
E/CameraService: checkTemperature (0)
E/CameraService: CameraService::connect X (PID 5316) rejected (existing client(s) with higher priority).
E/CameraService:    Conflicts with: Device 1, client package com.xxxx.xxxxx.debug (PID 16221, score 0, state 2)
E/SemCamera-JNI-Cpp: SemCamera_native_setup fail 7
```

COMMENTS

All comments ↓ Oldest first

hu...@google.com <hu...@google.com> #2 Apr 24, 2020 10:58AM

Reassigned to hu...@google.com.

Hi,

The issue you're seeing is due to the fact that `SurfaceView`'s surface becomes invalid once the lifecycle it's attached to gets stopped (for example, the `onStop()` callback in an `Activity` or `Fragment`). When the lifecycle is restarted, the camera requests a new valid surface, which only becomes available after `SurfaceView`'s `surfaceCreated()` callback, at that point a new capture request is made, and at some point after that the preview starts. This process may be the cause of the flickering you're noticing. This doesn't happen when using a `TextureView` because its surface stays valid, so a new capture request isn't made after the stop/start cycle.

[Deleted User] <[Deleted User]> #3 Apr 24, 2020 04:30PM

Thanks for the clarification,

So there should be some workaround to this, how do I prevent showing that gap? I've tried setting solid background colors but that gap is always transparent. Can you suggest something?

[Deleted User] <[Deleted User]> #4 May 1, 2020 06:30PM

Hi, any update on this?

hu...@google.com <hu...@google.com> #5 May 21, 2020 06:45AM

Hi,

Sorry for the late reply.

I've been trying to reproduce your issue, but wasn't able to so far. I constantly go back and forth between ActivityA and ActivityB, where ActivityB displays a camera preview using `PreviewView` with a `SurfaceView`. There doesn't seem to be any flickering though, only a short delay from the moment the activity is displayed to when the preview starts.

Could you please share a minimal code snippet that causes the issue you described? Also, is there a specific set of devices you're seeing this on? I've tested on a couple of Pixel phones, and a OnePlus 5T.

[Deleted User] <[Deleted User]> #6 May 21, 2020 02:10PM

Hi, thanks for the response.

We are testing a camera fragment placed inside an activity and when we try to open the activity, this flickering occurs:

Here's how we have attached the fragment:

```
val fragment = CameraFragment.newInstance()
        supportFragmentManager.beginTransaction()
            .setCustomAnimations(0, 0)
            .replace(R.id.fragment_container, fragment, fragment.javaClass.simpleName)
            .commitAllowingStateLoss()
```

Here's the code binding the Use cases:

```
val metrics = DisplayMetrics().also { viewFinder?.display?.getRealMetrics(it) }
Log.d(TAG, "Screen metrics: ${metrics.widthPixels} x ${metrics.heightPixels}")

val screenAspectRatio = aspectRatio(metrics.widthPixels, metrics.heightPixels)

val rotationConst = viewFinder?.display?.rotation

val cameraSelector = CameraSelector.Builder()
      .requireLensFacing(lensFacing)
      .build()
val cameraProviderFuture =
      ProcessCameraProvider.getInstance(requireContext())
cameraProviderFuture.addListener(Runnable {

   val cameraProvider: ProcessCameraProvider = cameraProviderFuture.get()

   preview = Preview.Builder()
         .setTargetAspectRatio(screenAspectRatio)
         .setTargetRotation(rotationConst ?: Surface.ROTATION_0)
         .build()

   imageCapture = ImageCapture.Builder()
         .setCaptureMode(ImageCapture.CAPTURE_MODE_MINIMIZE_LATENCY)
         .setTargetAspectRatio(screenAspectRatio)
         .setTargetRotation(Surface.ROTATION_0)
         .build()

   cameraProvider.unbindAll()

   try {
      val useCases = arrayOf(preview, imageCapture)
            .filterNotNull()
            .toTypedArray()

      val camera = cameraProvider.bindToLifecycle(this, cameraSelector, *useCases)

      viewFinder?.preferredImplementationMode = PreviewView.ImplementationMode.SURFACE_VIEW
      preview?.setSurfaceProvider(viewFinder?.createSurfaceProvider(camera.cameraInfo))
   }  catch (exc: Exception) {
      Log.e(TAG, "Some exception in the use cases : $exc")
   }
}, ContextCompat.getMainExecutor(requireContext()))
```

I don't know how it happens but this makes me easy to replicate the bug (5/10):
Launch the camera activity on tap of a button > Camera opens and preview keeps running > Quickly press back button to exit the camera activity > Without much delay tap on the button hard to open Camera again > Repeat.

Replicated on devices:
Asus Zenfone max pro m1, Samsung a70s, Samsung Note 10+, etc

I've attached a sample video that shows the flickering part (00:07-00:08).

VID-20200421-WA0002.mp4
3.2 MB Download

hu...@google.com <hu...@google.com> #7 Jun 12, 2020 10:17AM

Hi,

Is the flickering around second 7 from the video what you're referring to? If so, it seems that for a split of second the previous screen is visible before the preview starts, this may be because `PreviewView` is using a `SurfaceView`. You shouldn't see this issue on the latest version of camera-view, `PreviewView` now has a background color by default to prevent content behind the preview to show before the preview stream starts.

Could you please update to the latest version and verify whether this issue has been fixed? If you're still seeing the issue, could you set the preferred implementation mode to `TEXTURE_VIEW` and see if the issue is reproducible?

[Deleted User] <[Deleted User]> #8 Jun 12, 2020 09:22PM

Hi,

Thanks for your response. Yes, the flickering was around the second 7.
As we upgraded to 1.0.0-alpha11, the flickering is now not reproducible. However, for now, we would be still relying on TEXTURE_VIEW since, on some low-end phones, there are issues being reported with the use of SURFACE_VIEW.

Thanks for your assistance in resolving this.

hu...@google.com <hu...@google.com> #9 Jun 13, 2020 02:55AM

Glad to hear the issue's now resolved.

Could you please tell us more (Ideally by creating a new issue) about the issues you're seeing with SURFACE_VIEW on low end phones, and what some of these devices are?

---

hu...@google.com <hu...@google.com> Jun 24, 2020 02:12AM

Marked as fixed.