mCam

Android application manual

Contents

Introduction

mCam is an android camera application written in Java. It uses **`android.hardware.camera2`** API (level: 22). It allows users to manually control various parameters of phone camera like exposure time, frame rate, lens filter density etc. that were earlier automatically set by android. JPEG Metadata like GPS and values of sensors like gyroscope, accelerometer etc. can also enabled to be read at the time of capture. More control and precision is obtained by manual capture.

Application has modular design. It has kind of it's own API that has different functions to set parameters. It can be integrated with any controlling program to do different kind of jobs like capturing photos at regular interval, multiple phone capture etc.

Structure of application

To show how manual parameters can be controlled from outside program, application is built in two layers (also called activity) that interact with each other:

- *Launcher*
  It is a input screen that displays all controllable parameters for a phone camera. User can adjust parameters according to his needs and launch camera.
- *MainActivity*
  It shows the camera preview with all parameters received from launcher activity. User can click photos and save to SD card.

Screenshots to be attached here.

Working mechanism

In this new camera2 API, camera device is modelled as a pipeline, which takes in input requests for capturing a single frame, captures the single image per the request, and then outputs one capture result

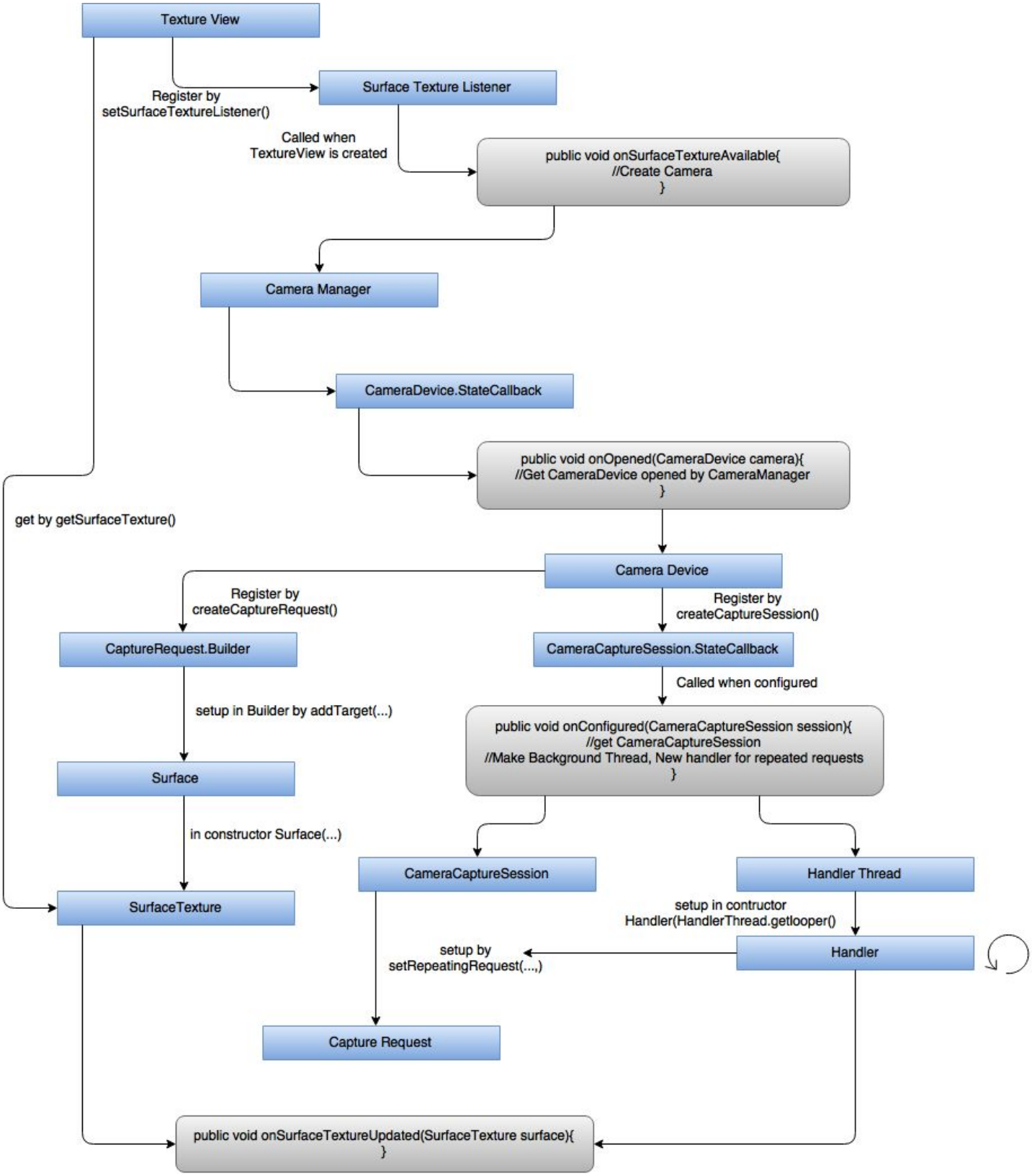
metadata packet, plus a set of output image buffers for the request.
Below is the step-by-step mechanism:

1. Launcher activity is created. It opens up the camera (without showing preview) and get all its characteristics. ‘Launch’ button listener is set up.
2. Once we have all the characteristics of a camera, UI screen shows the available values of focal lengths, filter densities and aperture in drop down list (called spinner). If there is no value supported for a given parameter, drop list shows ‘No Values’.
3. User select values from spinners and adjust exposure time, sensitivity and frame duration from seekbars.
4. When ‘Launch’ button is clicked, it launches MainActivity and sends all the parameter values.
5. MainActivity is created. All the parameters values sent by launcher activity are stored and preview is shown.

Preview mechanism
This section explains in detail the working of camera preview as per Camera2 API.
We need to set up following objects:

- CameraManager
    - Select Camera
    - Create CameraDevice
- CameraDevice
    - Create CaptureRequest
    - Create CameraCaptureSession
- CaptureRequest, CaptureRequest.CameraBuilder
    - Link Surface for Viewing
    - Make CaptureRequest *(contains all manual parameters)*
- CameraCaptureSession
    - Capture Camera Image and put the result on the Surface registered in CaptureRequest.

CAMERA PREVIEW FLOWCHART



API

It is basic set of functions written in Java upon which android camera application is built.

`public CaptureRequest.Builder` **`setSensitivity`**`(int value)`
Set sensor sensitivity to value.

`public CaptureRequest.Builder` **`setExposureTime`**`(CaptureRequest.Builder captureBuilder, int value)`
Set exposure time value.

`public CaptureRequest.Builder` **`setSensorFrameDuration`**`(CaptureRequest.Builder captureBuilder, int value)`
Set sensor frame duration value.

`public CaptureRequest.Builder` **`setFocusDistance`**`(CaptureRequest.Builder captureBuilder, int value)`
Set focus distance value.

`public CaptureRequest.Builder` **`setlensFilterDensity`**`(CaptureRequest.Builder captureBuilder,int value)`
Set lens filter density value.

`public CaptureRequest.Builder` **`setLensAperture`**`(CaptureRequest.Builder captureBuilder, int value)`
Set lens aperture value.

`public float[]` **`onGyroscopeChanged`**`(SensorEvent evt)`
Returns the sensor data array.

`public float[]` **`onAccelerometerChanged`**`(SensorEvent evt)`
Returns the sensor data array.

`public CaptureRequest.Builder` **`setJPEG_GPS_Location`**`(CaptureRequest.Builder captureBuilder)`
Enables or disables the GPS Metadata to be read.

`public parameterBundle` **`takePicture`**`(sensitivityValue, exposureTimeValue, sensorFrameDurationValue, focusDistanceValue, lensFilterDensityValue, lensAperture, GPSflag, gyroscopeFlag, accelerometerFlag)`
Capture the image and save it with all parameters.

`public CaptureRequest.Builder` **`setFlash`**`(CaptureRequest.Builder captureBuilder, int value)`
Based on value, we set the flash as On, Off or Torch.

References

https://developer.android.com/reference/android/hardware/camera2/package-summary.html