1. Thuộc tính android:alpha dùng để set độ trong suốt chỉ có thể dùng cho view, còn ko thể dùng cho drawable ☹

2. Để tạo 1 mảng lưu các giá trị resource id của các drawable, ta dùng arrays.xml trong values như sau:

Arrays.xml

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <string-array name="tabwidget\_icon">  
 <item>@drawable/icon\_home\_white</item>  
 <item>@drawable/icon\_search\_white</item>  
 <item>@drawable/icon\_camera</item>  
 <item>@drawable/icon\_news\_white</item>  
 <item>@drawable/icon\_profile\_white</item>  
 </string-array>  
</resources>

Ở trong code, muốn lấy resource id ta viết như sau:

TypedArray typedArray = getResources().obtainTypedArray(R.array.*tabwidget\_icon*);  
int size = typedArray.length();  
drawables = new Drawable[size];  
for (int i = 0; i < size; i++) {  
 drawables[i] = getResources().getDrawable(typedArray.getResourceId(i, -1));  
}

3. Để tạo một hình chữ nhật có 1 thanh gạch ngang phía dưới, ta sử dụng layer-list trong drawable như sau:

<?xml version="1.0" encoding="utf-8"?>  
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">  
 <item  
 android:top="-6dp"  
 android:left="-6dp"  
 android:right="-6dp"  
 android:bottom="0dp">  
  
 <shape android:shape="rectangle">  
 <solid android:color="#000000"/>  
 <stroke  
 android:width="5dp"  
 android:color="#125688"/>  
 </shape>  
 </item>  
</layer-list>

4. Tạo 1 Activity mở camera ngay trong app của mình

Các bước tạo ra 1 ứng dụng camera:

- **Detect and Access Camera:** code để kiểm tra máy có camera hay ko, sau đó viết access request truy cập đến camera.

- **Create a Preview Class:** Tạo 1 class preview camera thừa kế từ class SurfaceView and implements the **SurfaceHolder** interface. This class previews the live images from the camera.

- **Build a Preview Layout**: Once you have the camera preview class, create a view layout that incorporates(kết hợp) the preview and the user interface controls you want.

- **Setup Listeners for Capture**: Connect listeners for your interface controls to start image or video capture in response to user actions, such as pressing a button.

- **Capture and Save Files**: Setup the code for capturing pictures or videos and saving the output.

- **Release the Camera**: After using the camera, your application must properly release it for use by other applications.

//Đang bị lỗi vì sao hàm onResume lại chạy 2 lần. Nguyên nhân phỏng đoán là do đã set 2 Intent trong camera button.

Solved: Đã sửa đc bằng cách chỉ set 1 intent còn tabcontent set 1 layout rỗng.

//26/6 đang bị lỗi chỗ xoay ảnh trước khi lưu, nếu xoay ảnh thì ảnh gốc là 640 x 480 sẽ chuyển thành 480 x 640 và mất dữ liệu gốc nên khi lưu thành file ko đc. Nếu đổi width thành height trong hàm createBitmap thì sẽ bị lỗi: *IllegalArgumentException: y + height must be <= bitmap.height()*

Đã giải quyết được bằng cách viết hàm quay cho bức ảnh

//*29/6 đang bị lỗi chỗ màn hình preview, preview size khác với kích thước của surface view làm cho ảnh bị stretch.*

Thanh trạng thái chỉ giờ, tình trạng wifi trên đt được gọi là statusbar. Để lấy height của statusbar từ đó xác định size chính xác cho các layout trong đt ở trong code thay vì trong XML thì ta cần lấy height của status, cách lấy:

Đã sửa được cho ảnh khỏi stretch bằng cách tạo khung preview size đúng với các kích thước supported preview size của camera. Và chọn đúng aspect ratio tương ứng để giữ tỉ lệ 1:1

Còn chỗ ảnh trên gallery của đt hiển thị width x height = 3264 x 2448 là do cái gallery của Asus hiển thị bị sai, khi cop ảnh ra máy tính thì ta sẽ thu đc width x height = 2448 x 3264. Cái này mới chính xác.

//*30/6 đang tìm cách get optimal preview size and picture size.*

Đã xong, lỗi do ta truyền ko đúng w và h vào surface changed, lưu ý w > h

1/7 đã hoàn thành xong việc sử dụng OrientationEventListener để xác đinh hướng của thiết bị, từ đó quay ảnh lại đúng theo hướng yêu cầu.

//*Để import 1 project như là 1 module trong android studio*

*Ta chọn: New -> Import Module -> Chọn đên module đó*

*Nếu bị lỗi*

*Error:No such property: GROUP for class: org.gradle.api.publication.maven.internal.ant.DefaultGroovyMavenDeployer*

*Xóa dòng này trong file build.gradle của library mà ta vừa mới import vào:*

apply from: 'gradle-mvn-push.gradle'

Problem: Chú ý trong recycler view item, nếu ta vẽ đường thẳng bằng text view,

mà ko set text cho nó thì recycler view sẽ tự động xóa text view,

mặc dù text view ko có text trong các layout vẫn hiển thị bình thường

mà ko bị hủy. Dùng ImageView cũng ko đc, cho dù trong XML ta đã width = match\_parent nhưng khi hiển thị nó vẫn bị thu nhỏ lại chỉ còn 1 vạch. Đó là trường hợp set src, còn setbackground thì bị mất hoàn toàn.

Solution: set width trong code. Ta cho 1 thanh phan do width co dinh, xong do set width cua image view bang width cua thanh phan co dinh do dua vao LayoutParams.

//*Chú ý: Recycler View ko hỗ trợ cho chúng ta việc chọn từng item trong nó, mà chúng ta phải tự viết. Tham khảo cách lám selection handling trong Recycler View tại đây:*

[*http://enoent.fr/blog/2015/01/18/recyclerview-basics/*](http://enoent.fr/blog/2015/01/18/recyclerview-basics/)

Problem:

- Trong Android, khi ta đọc 1 file ảnh thành 1 mảng byte thì thứ tự màu sẽ là RGB thày vì BGR như trong java. Ảnh lưu trong Android là 32 bit, tương ứng 4 byte 1 pixel

- Khi ta chụp ảnh bằng camera, thì dữ liệu sẽ lưu dưới dạng 1 mảng byte[] data đặc biệt, không phải là 1 mảng byte các color channel như khi xử lý ảnh trong java.

Solution:

- Để lấy được mảng byte các channel color như trong xử lý ảnh ở Java, chúng ta chuyển mảng data thu được ở camera thành bitmap:

Bitmap bmp = BitmapFactory.decodeByteArray(data, 0, data.length);

Sau đó chuyển bitmap này thành mảng byte các màu, sử dụng bytebuffer, lưu ý ko sử dụng Bitmap.Compress như 1 số hướng dẫn trên mạng vì cách này sẽ cho ta mảng byte đã đc mã hóa theo 1 cách nào đó (cái này chưa biết ☺)

+ Phương pháp sai:

Bitmap bmp = intent.getExtras().get("data");

ByteArrayOutputStream stream = new ByteArrayOutputStream();

bmp.compress(Bitmap.CompressFormat.PNG, 100, stream);

byte[] byteArray = stream.toByteArray();

+Phương pháp đúng: Sử dụng ByteBuffer

ByteBuffer buffer = ByteBuffer.allocate(bmp.getByteCount());

bmp.copyPixelsToBuffer(buffer);

byte[] arra = buffer.array();

Nó sẽ trả về 1 mảng byte các màu theo thứ tự là RGBA: he so alpha duoc chuyen ra cuoi cung.

Sau đó nếu muốn chuyển mảng byte các màu này thành Bitmap như ban đầu thì ta cũng sử dụng byte buffer:

Buffer buffer1 = ByteBuffer.wrap(arra);

Bitmap newbmp = Bitmap.createBitmap(bmp.getWidth(), bmp.getHeight(), Bitmap.Config.ARGB\_8888);

newbmp.copyPixelsFromBuffer(buffer1);