

Mobile Application Development Camera

Waterford Institute of Technology

November 22, 2016

John Fitzgerald

Android Development

Developing camera module

Framework supports various cameras and camera features

Consider:

- Specify camera requirements in manifest
 - App only installable on device with camera?
- Still photo or video clip?
- Media available this app only or other?
 - Add to gallery?



Camera

Developer options

Two approaches available

- Use android camera API
 - Build customized app
- Use existing camera app
 - Use implicit intent
 - Similar to use of Maps app module in MyRent
 - Much simpler approach
 - But at a price



Camera

Building customized camera app (pre API 21)

Using deprecated **Camera API**

Superseded in API 21

- Detect & access camera
- Create preview class
- Build preview layout
- Setup and register listeners
- Capture and save files
- Release camera when done

```
/*interfaces*/  
SurfaceHolder.Callback,  
OnClickListener,  
Camera.ShutterCallback,  
Camera.PictureCallback  
/*classes*/  
Camera  
SurfaceView  
View
```

Camera

Building customized camera app (API 21)

package

android.hardware.camera2

Added in API 21

- Models camera as pipeline
- Input request to capture single frame
- Captures single image per request
- Outputs corresponding:
 - metadata packet
 - image buffers

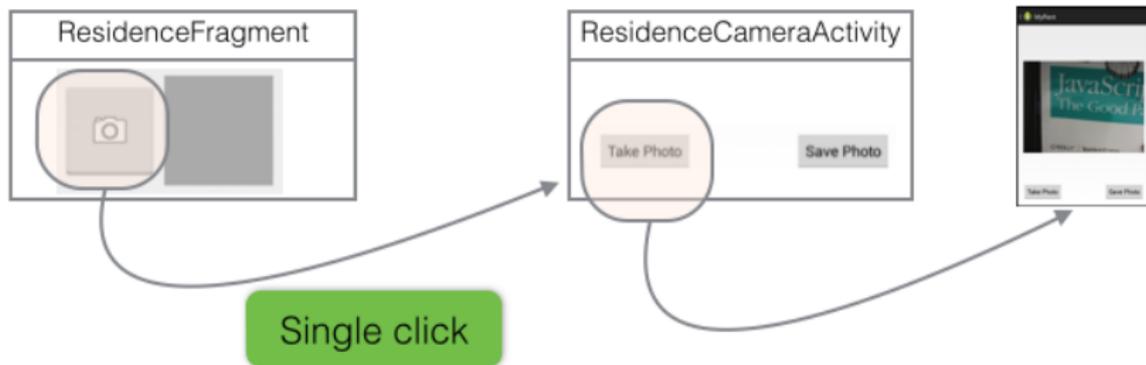
```
/** Example classes */  
CameraCaptureSession  
CameraCharacteristics  
CameraDevice  
CameraManager  
CameraMetadata  
CaptureFailure  
CaptureRequest  
CaptureResult  
CameraAccessException
```

Camera

Using existing camera app

Use implicit intent

- Click camera button to start process
- Switches to new activity
- Take photo
- Save photo
- Returned to ResidenceFragment
- Image displayed as thumbnail

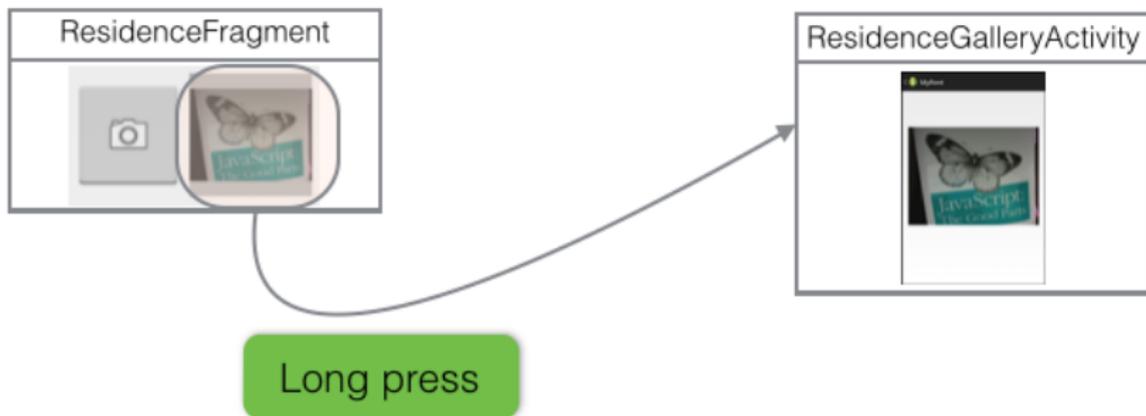


Camera

Using existing camera app

Display full-size photo

- Longpress thumbnail
- Switches to new activity
- Displays photo
- Use back or up button to return



Camera

Helpers

We begin with some helper methods

- Method to render photo ...
- ... that uses method to suitably scale image
 - *getScaledDrawable(Activity a, String path)*

```
/**  
 * Render the photo on the ImageView  
 */  
public static void showPhoto(Activity activity, Residence res, ImageView photoView)  
{  
    String path = activity.getFileStreamPath(res.photo).getAbsolutePath();  
    BitmapDrawable b = getScaledDrawable(activity, path);  
    if (b != null)  
        photoView.setImageDrawable(b);  
}
```

▶ photo = "da6676c8-2dd9-48b7-aaf9-40edfe84baf7.png" (id=831706398384)

Camera

Model

We will add the photo filename to the model layer

- A String object to ...
- ...store the photo filename

```
public String photo;  
  
public Residence()  
{  
    ...  
    photo = "photo";  
}
```

Camera

Model where file serialization used

Include the usual JSON to facilitate serialization

```
private static final String JSON_PHOTO = "photo" ;
public Residence(JSONObject json) throws JSONException
{
    ...
    photo = json.getString(JSON_PHOTO);
}
```

```
public JSONObject toJSON() throws JSONException
{
    ...
    json.put(JSON_PHOTO, photo);
    return json;
}
```

Camera

Photo-taking process

Begins in ResidenceFragment

- Widgets and listener

```
private ImageView cameraButton;  
private ImageView photoView;
```

```
private void addListeners(View v)  
{  
    ...  
    cameraButton = (ImageView) v.findViewById(R.id.camera_button);  
    ...  
    cameraButton.setOnClickListener(this);  
}
```

Camera

Photo-taking process

Listener response to camera button press

- start activity
- expect photo filename in result

```
@Override
public void onClick(View v)
{
    switch (v.getId())
    {
        ...
        case R.id.camera_button:
            Intent ic = new Intent(getActivity(), ResidenceCameraActivity.class);
            startActivityForResult(ic, REQUEST_PHOTO);
            break;
    }
}
```

Camera

Photo-taking process

On successful taking of photo

- photo filename returned
- update model with returned filename
- render photo as thumbnail

```
@Override
public void onActivityResult(int requestCode, int resultCode, Intent data)
{
    ...
    switch (requestCode)
    {
        ...
        case REQUEST_PHOTO:
            String filename = data.getStringExtra(ResidenceCameraActivity.EXTRA_PHOTO_FILENAME);
            if (filename != null) {
                residence.photo = filename;
                showPhoto(getActivity(), residence, photoView );
            }
            break;
    }
}
```

Camera

Display full-size photo

Set longpress listener

- ResidenceFragment implements *View.OnLongClickListener*

```
private void addListeners(View v)
{
    ...
    photoView = (ImageView) v.findViewById(R.id.myrent_imageView);
    ...
    photoView.setOnLongClickListener(this);
}
```

Camera

Display full-size photo

Respond to longpress

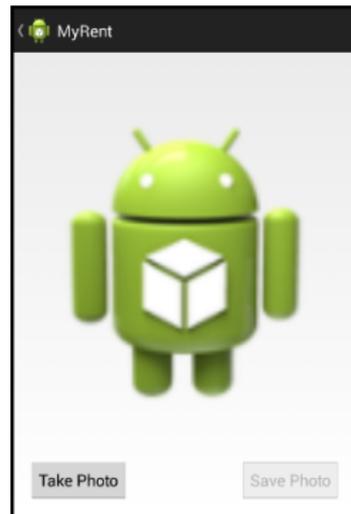
- Stores this Residence id in ResidenceGalleryActivity intent
- Switches to ResidenceGalleryActivity

```
@Override
public boolean onLongClick(View v)
{
    Intent i = new Intent(getActivity(), ResidenceGalleryActivity.class);
    i.putExtra(EXTRA_RESIDENCE_ID, residence.id);
    startActivity(i);
    return true;
}
```

Camera

ResidenceCameraActivity

- Activity layout comprises
 - placeholder for photo
 - button to take photo
 - button to save photo
- Save button disabled pending photo shoot
- Up button behaves same as back button
 - returns to calling activity



Camera

ResidenceCameraActivity: Take photo

- Widgets and listeners

```
private Button    savePhoto;
private Button    takePhoto;

@Override
protected void onCreate(Bundle savedInstanceState)
{
    savePhoto = (Button)findViewById(R.id.savePhoto);
    takePhoto = (Button)findViewById(R.id.takePhoto);
    savePhoto.setEnabled(false);
    savePhoto.setOnClickListener(this);
    takePhoto.setOnClickListener(this);
}
```

Camera

ResidenceCameraActivity: Take photo

- Program **Up** button to behave same as **Back** button

```
@Override
protected void onCreate(Bundle savedInstanceState)
{
    ...
    getActionBar().setDisplayHomeAsUpEnabled(true);
}
```

Camera

ResidenceCameraActivity: Take photo

- Program **Up** button to behave same as **Back** button

```
@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    switch (item.getItemId())
    {
        case android.R.id.home : onBackPressed();
                                return true;
        default                 : return super.onOptionsItemSelected(item);
    }
}
```

Camera

ResidenceCameraActivity: Take photo

- Use the device camera to capture image

```
@Override
public void onClick(View v)
{
    switch(v.getId())
    {
        case R.id.takePhoto      : onTakePhotoClicked(v);
                                break;
    }
}

public void onTakePhotoClicked(View v)
{
    Intent cameraIntent = new Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
    startActivityForResult(cameraIntent, CAMERA_RESULT);
    savePhoto.setEnabled(true);
}
```

Camera

ResidenceCameraActivity: Take photo

- Intent containing photo returned

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
    super.onActivityResult(requestCode, resultCode, data);
    switch (requestCode)
    {
        case ResidenceCameraActivity.CAMERA_RESULT : processImage(data);
                                                    break;
    }
}
```

Camera

ResidenceCameraActivity: Take photo

- Process returned image

```
private void processImage(Intent data)
{
    residencePhoto = (Bitmap) data.getExtras().get("data");
    if(residencePhoto == null)
    {
        Toast.makeText(this, "Photo shoot failed", Toast.LENGTH_SHORT).show();
    }
    residenceImage.setImageBitmap(residencePhoto);
}
```

Camera

ResidenceCameraActivity: Take photo

- Declare static constants

```
private static final int    CAMERA_RESULT = 5;  
public static final String EXTRA_PHOTO_FILENAME = "org.wit.myrent.photo.filename";
```

Camera

ResidenceCameraActivity: Save photo

- Widgets and listener

```
@Override
protected void onCreate(Bundle savedInstanceState)
{
    ...
    savePhoto = (Button)findViewById(R.id.savePhoto);
    savePhoto.setEnabled(false);
    savePhoto.setOnClickListener(this);
}
```

Camera

ResidenceCameraActivity: Save photo

- When photo taken then save button enabled

```
@Override
public void onClick(View v)
{
    switch(v.getId())
    {
        ...
        case R.id.savePhoto : onPictureTaken(residencePhoto);
                             break;
    }
}
```

Camera

ResidenceCameraActivity: Save photo

- Create random filename
- Use helper method to save photo
- Put filename in intent & **setResult** to return to calling activity

```
private void onPictureTaken(Bitmap data)
{
    String filename = UUID.randomUUID().toString() + ".png";
    if(writeBitmap(this, filename, data) == true) {
        Intent intent = new Intent();
        intent.putExtra(EXTRA_PHOTO_FILENAME, filename);
        setResult(Activity.RESULT_OK, intent);
    }
    else {
        setResult(Activity.RESULT_CANCELED);
    }
    finish();
}
```

Camera

ResidenceFragment

- Retrieve photo filename from intent
- Update the model
- Invoke helper *showPhoto* to render image as thumbnail

```
@Override
public void onActivityResult(int requestCode, int resultCode, Intent data)
{
    ...
    switch (requestCode)
    {
        ...
        case REQUEST_PHOTO:
            String filename = data.getStringExtra(ResidenceCameraActivity.EXTRA_PHOTO_FILENAME);
            if (filename != null) {
                residence.photo = filename;
                showPhoto(getActivity(), residence, photoView );
            }
            break;
    }
}
```

Camera

CameraHelper

- Recall that photo already saved to device
- Obtain absolute path
- Render bitmap (scaled using helper)

```
/**
 * Render the photo on the ImageView
 */
public static void showPhoto(Activity activity, Residence res, ImageView photoView)
{
    String path = activity.getFileStreamPath(res.photo).getAbsolutePath();
    BitmapDrawable b = getScaledDrawable(activity, path);
    if (b != null)
        photoView.setImageDrawable(b);
}
```

Camera

Gallery

- Longpress thumbnail in ResidenceFragment
- Switches to ResidenceGalleryActivity
 - ImageView widget declared: this to contain photo
 - Inflate xml layout: image placeholder
 - Invoke *showPicture* in *onCreate*

```
private ImageView photoView;  
  
photoView = (ImageView) findViewById(R.id.residenceGalleryImage);  
  
showPicture();
```

Camera

Gallery: Show picture

- Obtain Residence UUID from intent
- Use to get residence referred to
- Invoke helper *showPhoto* to render image
- Return using Up or Back buttons

```
private void showPicture()
{
    UUID resId = (UUID) getIntent().getSerializableExtra(ResidenceFragment.EXTRA_RESIDENCE_ID);
    MyRentApp app = (MyRentApp) getApplication();
    Portfolio portfolio = app.portfolio;
    Residence residence = portfolio.getResidence(resId);
    showPhoto(this, residence, photoView);
}
```