

## Ćwiczenia 13 — Android studio – CameraX + zapis na sd + zapis do galerii + video

Na koniec zajęć prześlij pliki źródłowe (.xml, .java)+ obrazek do zasobu w teams.

1. Wzorujemy się na labie: <https://developer.android.com/codelabs/camerax-getting-started#0>
2. Utwórz projekt o nazwie CameraX na podstawie Empty Activity, dobierz odpowiednie API ( ).
3. Otworzyć dokumentację:

<https://developer.android.com/training/camerax>

<https://developer.android.com/training/data-storage>

<https://developer.android.com/reference/android/provider/MediaStore#summary>

<https://developer.android.com/reference/android/os/Environment>

4. Dodaj zależności w build.gradle(Module: CameraX.app)

```
compileOptions {  
    sourceCompatibility JavaVersion.VERSION_1_8  
    targetCompatibility JavaVersion.VERSION_1_8  
}  
  
dependencies {  
    // CameraX core library using camera2 implementation  
    implementation "androidx.camera:camera-camera2:1.0.2"  
    // CameraX Lifecycle Library  
    implementation "androidx.camera:camera-lifecycle:1.0.2"  
    // CameraX View class  
    implementation 'androidx.camera:camera-view:1.0.0-alpha31'  
    //  
}
```

5. Dodaj w xml Button, PreviewView w dowolnym layout, np.

```
1      <?xml version="1.0" encoding="utf-8"?>
2      <androidx.constraintlayout.widget.ConstraintLayout
3          xmlns:android="http://schemas.android.com/apk/res/android"
4          xmlns:tools="http://schemas.android.com/tools"
5          xmlns:app="http://schemas.android.com/apk/res-auto"
6          android:layout_width="match_parent"
7          android:layout_height="match_parent"
8          tools:context=".MainActivity">
9
10         <Button
11             android:id="@+id/camera_capture_button"
12             android:layout_width="100dp"
13             android:layout_height="100dp"
14             android:layout_marginBottom="50dp"
15             android:scaleType="fitCenter"
16             android:text="Take Photo"
17             app:layout_constraintLeft_toLeftOf="parent"
18             app:layout_constraintRight_toRightOf="parent"
19             app:layout_constraintBottom_toBottomOf="parent"
20             android:elevation="2dp" />
21
22         <androidx.camera.view.PreviewView
23             android:id="@+id/previewView"
24             android:layout_width="match_parent"
25             android:layout_height="match_parent" />
26
27     </androidx.constraintlayout.widget.ConstraintLayout>
```

## 6. Uzupełnij MainActivity.java:

```

public class MainActivity extends AppCompatActivity implements CameraXConfig.Provider{
    private ListenableFuture<ProcessCameraProvider> cameraProviderFuture;
    private static final int MY_CAMERA_PERMISSION_CODE = 2;
    private ImageCapture imageCapture = null;
    private Camera camera;
    private File outputDirectory;
    private ExecutorService cameraExecutor;
    Button camera_capture_button;
    private final String TAG = "CameraXBasic";
    private final String FILENAME_FORMAT = "yyyy-MM-dd-HH-mm-ss-SSS";
    private final int REQUEST_CODE_PERMISSIONS = 10;
    private String [] REQUIRED_PERMISSIONS = new String[]{"android.permission.CAMERA"};
}
/**
 * https://developer.android.com/training/camerax
 * https://developer.android.com/codelabs/camerax-getting-started#0
 */
PreviewView previewView;
@Override
protected void onCreate(Bundle savedInstanceState) {

```

## 7. Dalej, szkielelet dla onCreate():

```

56 protected void onCreate(Bundle savedInstanceState) {
57     super.onCreate(savedInstanceState);
58     setContentView(R.layout.activity_main);
59     camera_capture_button = findViewById(R.id.camera_capture_button);
60     previewView = findViewById(R.id.previewView);
61     //
62
63     // Request camera permissions
64     if (allPermissionsGranted()) {
65         startCamera();
66     } else {
67         ActivityCompat.requestPermissions( activity: this, REQUIRED_PERMISSIONS, REQUEST_CODE_PERMISSIONS);
68     }
69
70     // Set up the listener for take photo button
71     camera_capture_button.setOnClickListener(view ->
72         takePhoto()
73     );
74
75     outputDirectory = getOutputDirectory();
76
77     cameraExecutor = Executors.newSingleThreadExecutor();
78
79 }

```

```
185 private boolean allPermissionsGranted() {
186     if (getApplicationContext().checkSelfPermission(Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED)
187     {
188         requestPermissions(new String[]{Manifest.permission.CAMERA}, MY_CAMERA_PERMISSION_CODE);
189     }
190     return true;
}
```

8. Dodaj i uzupełnij metodę `getOutputDirectory()` - rozbuduj ją :

```
private void startCamera(){ }

private File getOutputDirectory(){
    return getExternalFilesDir(Environment.DIRECTORY_PICTURES);
}

@Override
protected void onDestroy() {
    super.onDestroy();
    executorService.shutdown();
}
```

9. Dodaj do `AndroidManifest.xml`

```
activity_main.xml x MainActivity.java x build.gradle (:app) x AndroidManifest.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.camerax">
4
5     <uses-feature android:name="android.hardware.camera.any" />
6     <uses-permission android:name="android.permission.CAMERA" />
7     <application
8         android:allowBackup="true"
9         android:icon="@mipmap/ic_launcher"
10        android:label="CameraX"
11        android:roundIcon="@mipmap/ic_launcher_round"
```

10. Dodaj kolejną metodę po Ctrl+o:

```
168      @Override
169      public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
170          super.onRequestPermissionsResult(requestCode, permissions, grantResults);
171          if (requestCode == MY_CAMERA_PERMISSION_CODE)
172          {
173              if (grantResults[0] == PackageManager.PERMISSION_GRANTED)
174              {
175                  Toast.makeText(getApplicationContext(), text: "camera permission granted", Toast.LENGTH_LONG).show();
176                  startCamera();
177              }
178              else
179              {
180                  Toast.makeText(getApplicationContext(), text: "camera permission denied", Toast.LENGTH_LONG).show();
181                  finish();
182              }
183          }
184      }
```

11. Dodaj metodę startCamera:

```
131 private void startCamera() {
132     cameraProviderFuture = ProcessCameraProvider.getInstance(context: this);
133     cameraProviderFuture.addListener(() -> {
134         try {
135             ProcessCameraProvider cameraProvider = cameraProviderFuture.get();
136             // bindPreview(cameraProvider);
137             Preview preview = new Preview.Builder()
138                 .build();
139
140             CameraSelector cameraSelector = new CameraSelector.Builder()
141                 .requireLensFacing(CameraSelector.LENS_FACING_BACK)
142                 .build();
143
144             preview.setSurfaceProvider(previewView.getSurfaceProvider());
145
146             imageCapture = new ImageCapture.Builder()
147                 .build();
148             Log.v(TAG, msg: "startCamera");
149             try {
150                 // Unbind use cases before rebinding
151                 cameraProvider.unbindAll();
152
153                 // Bind use cases to camera
154                 camera = cameraProvider.bindToLifecycle((LifecycleOwner) this, cameraSelector, preview, imageCapture);
155                 // cameraProvider.bindToLifecycle(lifecycleOwner, cameraSelector, preview, imageCapture);
156             } catch (Exception e) {
157                 Log.e(TAG, msg: "Use case binding failed 1" + e);
158             }
159
160         } catch (ExecutionException | InterruptedException e) {
161             // No errors need to be handled for this Future.
162             // This should never be reached.
163             Log.e(TAG, msg: "Use case binding failed" + e);
164         }
165     }, ContextCompat.getMainExecutor(context: this));
166
167 }
```

12. Dodaj zawartość takePhoto():

```
86 private void takePhoto() {
87     // Get a stable reference of the modifiable image capture use case
88     ImageCapture imageCapture = this.imageCapture;
89     // Create time-stamped output file to hold the image
90     File file = null;
91     try {
92         file = File.createTempFile(
93             new SimpleDateFormat(FILENAME_FORMAT, Locale.US
94             ).format(System.currentTimeMillis()), suffix: ".jpg", this.outputDirectory);
95         Log.v(TAG, msg: "nazwa pliku: "+file.toString());
96     } catch (IOException e) {
97         e.printStackTrace();
98     }
99     Log.v(TAG, msg: "start takePhoto()");
100    // Create output options object which contains file + metadata
101    File photoFile = file;
102    ImageCapture.OutputFileOptions outputFileOptions =
103        new ImageCapture.OutputFileOptions.Builder(photoFile).build();
104    Log.v(TAG, msg: "output"+outputFileOptions);
105
106    imageCapture.takePicture(outputFileOptions, ContextCompat.getMainExecutor((Context)this),
107        (new ImageCapture.OnImageSavedCallback() {
108
109            @Override
110            public void onImageSaved(ImageCapture.OutputFileResults outputFileResults) {
111                Uri savedUri = Uri.fromFile(photoFile);
112                String msg = "Photo capture succeeded: "+savedUri;
113                Toast.makeText(getApplicationContext(),msg,Toast.LENGTH_SHORT).show();
114                Log.d(TAG, msg);
115            }
116
117            @Override
118            public void onError(ImageCaptureException error) {
119                Log.e(TAG, msg: "Photo capture failed: "+error.getMessage());
120            }
121        }));
122 }
```

13. Reszta z dokumentacji dla klasy CameraX i: <https://developer.android.com/codelabs/camerax-getting-started#4>

14. Sprawdź, czy zdjęcie zapisuje się w pamięci wew.→Android→data→com.example.CameraX->files->Pictures

15. Dodaj zdjęcie do galerii.

16. Dodatkowe zadania

- dodaj obsługę zapisu na kartę SD
- przeprowadź zapis video do pliku mp4
- dodaj obsługę bokeh, night,:

<https://developer.android.com/training/camerax/vendor-extensions>

17. KONIEC.