WSTĘP DO ANDROIDA

Laboratorium 6
Systemy i aplikacje bez granic

- Tworzymy nowy projekt typu Empty Acvivity o nazwie PermissionExample

- Dodajemy pole do klasy

```
private val TAG = "PermissionDemo"
```

- Dodajemy do klasy metodę setupPermissions i jej wywołanie w metodzie onCreate

```
private fun setupPermissions() {
    val permission = ContextCompat.checkSelfPermission( context: this,
            Manifest.permission.RECORD_AUDIO)

    if (permission != PackageManager.PERMISSION_GRANTED) {
        Log.i(TAG,  msg: "Brak zgody na nagrywanie")
    }
    else  {
        Log.i(TAG,  msg: "Mam zgodę na nagrywanie")
    }
}
```

I

- Uruchamiamy i sprawdzamy LogCat
- Do manifestu dodajemy linijkę

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
```

- I uruchamiamy na emulatorze z systemem starszym niż 6 sprawdzając LogCat

- Uruchamiamy teraz na emulatorze z systemem >= 6 i śledzimy LogCat

- Dodajemy drugie pole do klasy

```
private val RECORD_REQUEST_CODE = 101
```

- I metodę makeRequest

```
if (permission != PackageManager.PERMISSION_GRANTED) {
    Log.i(TAG,  msg: "Brak zgody na nagrywanie")
    makeRequest()
}
```

```
private fun makeRequest() {
    ActivityCompat.requestPermissions( activity: this,
            arrayOf(Manifest.permission.RECORD_AUDIO),
            RECORD_REQUEST_CODE)
}
```

- Dodajemy drugą metodę - onRequestPermissionsResult

```
override fun onRequestPermissionsResult(requestCode: Int,
                                        permissions: Array<String>, grantResults: IntArray) {
    when (requestCode) {
        RECORD_REQUEST_CODE -> {

            if (grantResults.isEmpty() || grantResults[0] != PackageManager.PERMISSION_GRANTED) {

                Log.i(TAG,  msg: "Użytkownik nie wyraził zgody")
            } else {
                Log.i(TAG,  msg: "Użytkownik wyraził zgodę")
            }
        }
    }
}
```

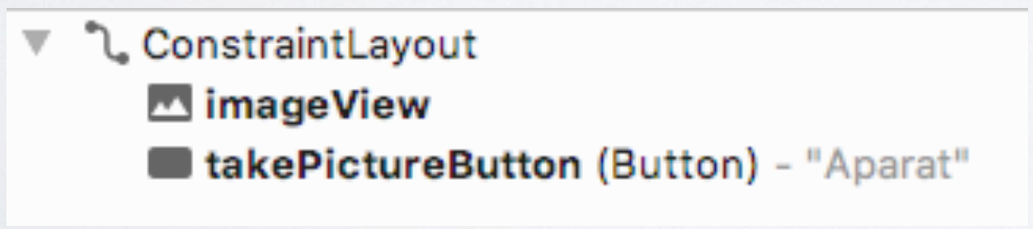
- Uruchamiamy i sprawdzamy na starym i nowym systemie.

- Sprawdzamy działanie zarządzania uprawnieniami w emulatorze

II

- Tworzymy nowy projekt typu Empty Activity o nazwie CameraExample

- Pośrodku aktywności umieszczamy ImageView,
  a poniżej Button



II

- Do manifestu dodajemy 3 linijki pozwoleń

```
<uses-permission-sdk-23 android:name="android.permission.CAMERA" />
<uses-permission-sdk-23 android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission-sdk-23 android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

- Do metody onCreate dodajemy

```
if (ContextCompat.checkSelfPermission( context: this, Manifest.permission.CAMERA) !=
        PackageManager.PERMISSION_GRANTED) {
    takePictureButton.setEnabled(false)
    ActivityCompat.requestPermissions( activity: this, arrayOf(Manifest.permission.CAMERA,
            Manifest.permission.WRITE_EXTERNAL_STORAGE), requestCode: 0)
}
```

II

- Oraz metodę

```
override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<String>, grantResults: IntArray) {
    if (requestCode == 0) {
        if (grantResults.size > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED
            && grantResults[1] == PackageManager.PERMISSION_GRANTED) {
            takePictureButton?.setEnabled(true)
        }
    }
}
```

- I metodę obsługującą przycisk

```
fun takePicture(v:View) {
    val intent = Intent(MediaStore.ACTION_IMAGE_CAPTURE)
    startActivityForResult(intent, requestCode: 100)
}
```

II

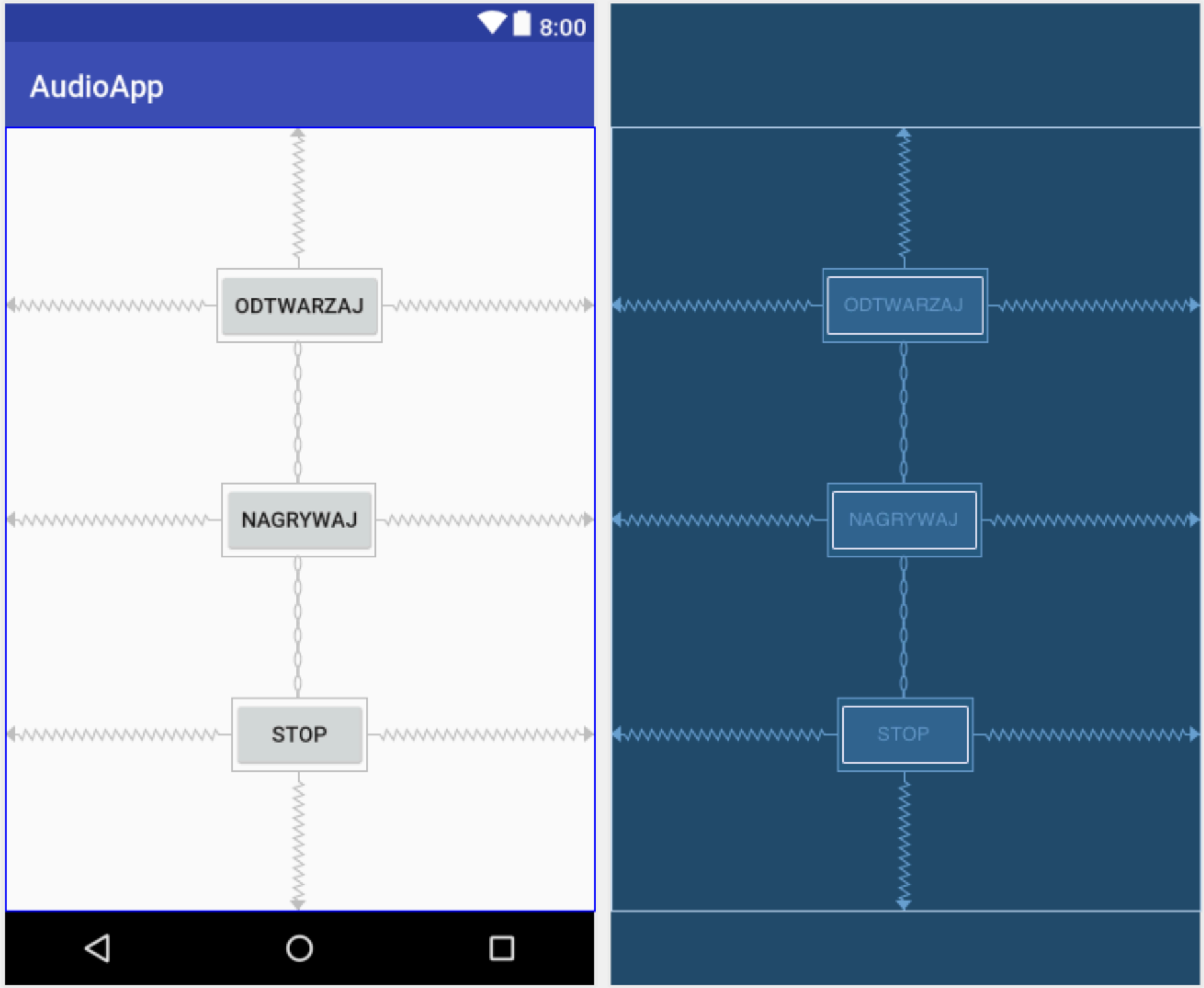
- Dodajemy też powrót z aktywności

```
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    if (resultCode== Activity.RESULT_OK) {
        if (requestCode==100) {
            if (data?.data != null) {
            } else {
                val image = data?.extras?.get("data") as? Bitmap
                imageView?.setImageBitmap(image)
            }

        }
    }
}
```

III

- Tworzymy nowy projekt typu Empty Activity o nazwie AudioApp
- Usuwamy TextView i umieszczamy 3 przyciski w kolumnie: Odtwarzaj, Nagraj, Stop - tworzące Vertical Chain - nazwane playButton, recordButton i stopButton


8:00
AudioApp
ODTWARZAJ
NAGRYWAJ
STOP
ODTWARZAJ
NAGRYWAJ
STOP

III

- Dodajemy do klasy pola

```
private var mediaRecorder: MediaRecorder? = null
private var mediaPlayer: MediaPlayer? = null
private var audioFilePath: String? = null
private var isRecording = false

private val RECORD_REQUEST_CODE = 101
private val STORAGE_REQUEST_CODE = 102
```

- I metodę sprawdzającą, czy urządzenie ma mikrofon

```
private fun hasMicrophone(): Boolean {
    val pmanager = this.packageManager
    return pmanager.hasSystemFeature(
            PackageManager.FEATURE_MICROPHONE)
}
```

III

- W manifeście dodajemy pozwolenia

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
```

- I obsługę pozwoleń w kodzie

```
private fun requestPermission(permissionType: String, requestCode: Int) {
    val permission = ContextCompat.checkSelfPermission( context: this,
            permissionType)

    if (permission != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions( activity: this,
                arrayOf(permissionType), requestCode
        )
    }
}
```

```
override fun onRequestPermissionsResult(requestCode: Int,
                                        permissions: Array<String>, grantResults: IntArray) {
    when (requestCode) {
        RECORD_REQUEST_CODE -> {

            if (grantResults.isEmpty() || grantResults[0]
                    != PackageManager.PERMISSION_GRANTED) {

                recordButton.isEnabled = false

                Toast.makeText( context: this,
                        text: "Wymagana zgoda na nagrywanie",
                        Toast.LENGTH_LONG).show()
            } else {
                requestPermission(
                        Manifest.permission.WRITE_EXTERNAL_STORAGE,
                        STORAGE_REQUEST_CODE)
            }
            return
        }
        STORAGE_REQUEST_CODE -> {

            if (grantResults.isEmpty() || grantResults[0]
                    != PackageManager.PERMISSION_GRANTED) {
                recordButton.isEnabled = false
                Toast.makeText( context: this,
                        text: "Wymagana zgoda na zapis w pamięci zewnętrznej",
                        Toast.LENGTH_LONG).show()
            }
            return
        }
    }
}
```

III

- Dodajemy metodę audioSetup i jej wywołanie w onCreate

```
private fun audioSetup() {

    if (!hasMicrophone()) {
        stopButton.isEnabled = false
        playButton.isEnabled = false
        recordButton.isEnabled = false
    } else {
        playButton.isEnabled = false
        stopButton.isEnabled = false
    }

    audioFilePath = Environment.getExternalStorageDirectory()
            .absolutePath + "/myaudio.3gp"
    requestPermission(Manifest.permission.RECORD_AUDIO,
            RECORD_REQUEST_CODE)
}
```

III

- Oraz metody obsługi przycisków

```
fun recordAudio(view: View) {
    isRecording = true
    stopButton.isEnabled = true
    playButton.isEnabled = false
    recordButton.isEnabled = false

    try {
        mediaRecorder = MediaRecorder()
        mediaRecorder?.setAudioSource(MediaRecorder.AudioSource.MIC)
        mediaRecorder?.setOutputFormat(
                MediaRecorder.OutputFormat.THREE_GPP)
        mediaRecorder?.setOutputFile(audioFilePath)
        mediaRecorder?.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB)
        mediaRecorder?.prepare()
    } catch (e: Exception) {
        e.printStackTrace()
    }
    mediaRecorder?.start()
}
```

|||

```
fun stopAudio(view: View) {

    stopButton.isEnabled = false
    playButton.isEnabled = true

    if (isRecording) {
        recordButton.isEnabled = false
        mediaRecorder?.stop()
        mediaRecorder?.release()
        mediaRecorder = null
        isRecording = false
    } else {
        mediaPlayer?.release()
        mediaPlayer = null
        recordButton.isEnabled = true
    }
}
```

```
fun playAudio(view: View) {
    playButton.isEnabled = false
    recordButton.isEnabled = false
    stopButton.isEnabled = true

    mediaPlayer = MediaPlayer()
    mediaPlayer?.setDataSource(audioFilePath)
    mediaPlayer?.prepare()
    mediaPlayer?.start()
}
```