Ubiquitous and Mobile Computing
CS 528: Media Recorder & Speaking to Android

Boya Zhou, Chu Wang, Jin Huang,
Roger Wirkala, Yuhan Liu

Computer Science Dept.
Worcester Polytechnic Institute (WPI)
Android Media Recorder

- Mobile phones have been equipped with sensors able measure and record their surroundings since their inception.
- Media creation and consumption is constantly evolving and so is the way we interact with it.
Problem Solved

- Easy, intuitive, and streamlined API for developers to leverage the capture and consumption of various different media mediums
Typical Use Case

- Social Media
- Supplemental material
- Communication
- Almost endless creative possibilities
Real World Examples

- Instagram
  - Instagram
  - 

- Tumblr
  - Tumblr, Inc.

- Snapchat
  - Snap Inc

- Twitter
  - Twitter, Inc.

- Facebook
  - Facebook

- Media Recorder
  - Mobile Solutions World

- Voice Recorder
  - quality apps (recorder

- Screen Recorder
  - VideoShow EnjoyMobi

- Smart Video Record
  - INTank Corp

- Screen Recorder
  - AppSmartz
Overview of how it works

State machine of the recording control:

Initial state:
Initialize a new instance of MediaRecorder with the following calls

Prepare state:
Complete the initialization by calling prepare()

Released state:
When you are done with MediaRecorder instance free its resources as soon as possible by calling release()
Code snippet (create and run a MediaRecorder)

```java
MediaRecorder recorder = new MediaRecorder();
    recorder.setAudioSource(MediaRecorder.AudioSource.MIC);
recorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
recorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
recorder.setOutputFile(PATH_NAME);
recorder.start();  // recording is now started
...
recorder.stop();
recorder.reset();
recorder.release();  // Now the object cannot be reused
```
Overview of how it works

- To be able to record, your app must tell the user that it will access the device’s audio input. Include the following permission tag in the app’s manifest file:

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

- **RECORD_AUDIO** is considered a “dangerous” permission since it may pose risk to a user’s privacy. Starting with Android 6.0 an app that uses a dangerous permission must ask the user for approval at run time. `ActivityCompat.requestPermissions()` is used to implement this behavior.
public class AudioRecordTest extends AppCompatActivity {

    private static final String LOG_TAG = "AudioRecordTest";
    private static final int REQUEST_RECORD_AUDIO_PERMISSION = 200;

    // Requesting permission to RECORD_AUDIO
    private boolean permissionToRecordAccepted = false;
    private String [] permissions = {Manifest.permission.RECORD_AUDIO};

    @Override
    public void onRequestPermissionsResult(int requestCode,
            @NonNull String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        switch (requestCode){
            case REQUEST_RECORD_AUDIO_PERMISSION:
                permissionToRecordAccepted = grantResults[0] == PackageManager.PERMISSION_GRANTED;
                break;
        }
        if (!permissionToRecordAccepted ) finish();
    }
}
private void onRecord(boolean start) {
    if (start) {
        startRecording();
    } else {
        stopRecording();
    }
}

private void onPlay(boolean start) {
    if (start) {
        startPlaying();
    } else {
        stopPlaying();
    }
}

private void startPlaying() {
    mPlayer = new MediaPlayer();
    try {
        mPlayer.setDataSource(mFileName);
        mPlayer.prepare();
        mPlayer.start();
    } catch (IOException e) {
        Log.e(LOG_TAG, "prepare() failed");
    }
}
Code snippet (how to make video recording)

```java
private void startRecording() {
    mRecorder = new MediaRecorder();
    mRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
    mRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
    mRecorder.setOutputFile(mFileName);
    mRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);

    try {
        mRecorder.prepare();
    } catch (IOException e) {
        Log.e(LOG_TAG, "prepare() Failed");
    }

    mRecorder.start();
}

private void stopRecording() {
    mRecorder.stop();
    mRecorder.release();
    mRecorder = null;
}
```