CSc 317
AsyncTask, MediaPlayer, ConstraintLayout

Benjamin Dicken

Three ways of Multi-Threading

AsyncTask

Pro: Nice interface   Con: deprecated :(

Runnable + Thread

Pro: Standard Java   Con: Awkward to run on UI thread

Executor + Handler + Runnable + Looper

Pro: Some standard Java, Thread Pools

Con: Awkward to run on UI thread

https://stackoverflow.com/a/64969640

http://tutorials.jenkov.com/java-util-concurrent/executorservice.html

AsyncTask

- Enables proper and easy use of the UI thread
- Allows you to perform background operations and publish results on the UI thread without having to manipulate threads and/or handlers
- Should ideally be used for short operations
    - a few seconds at the most

https://developer.android.com/reference/android/os/AsyncTask.html

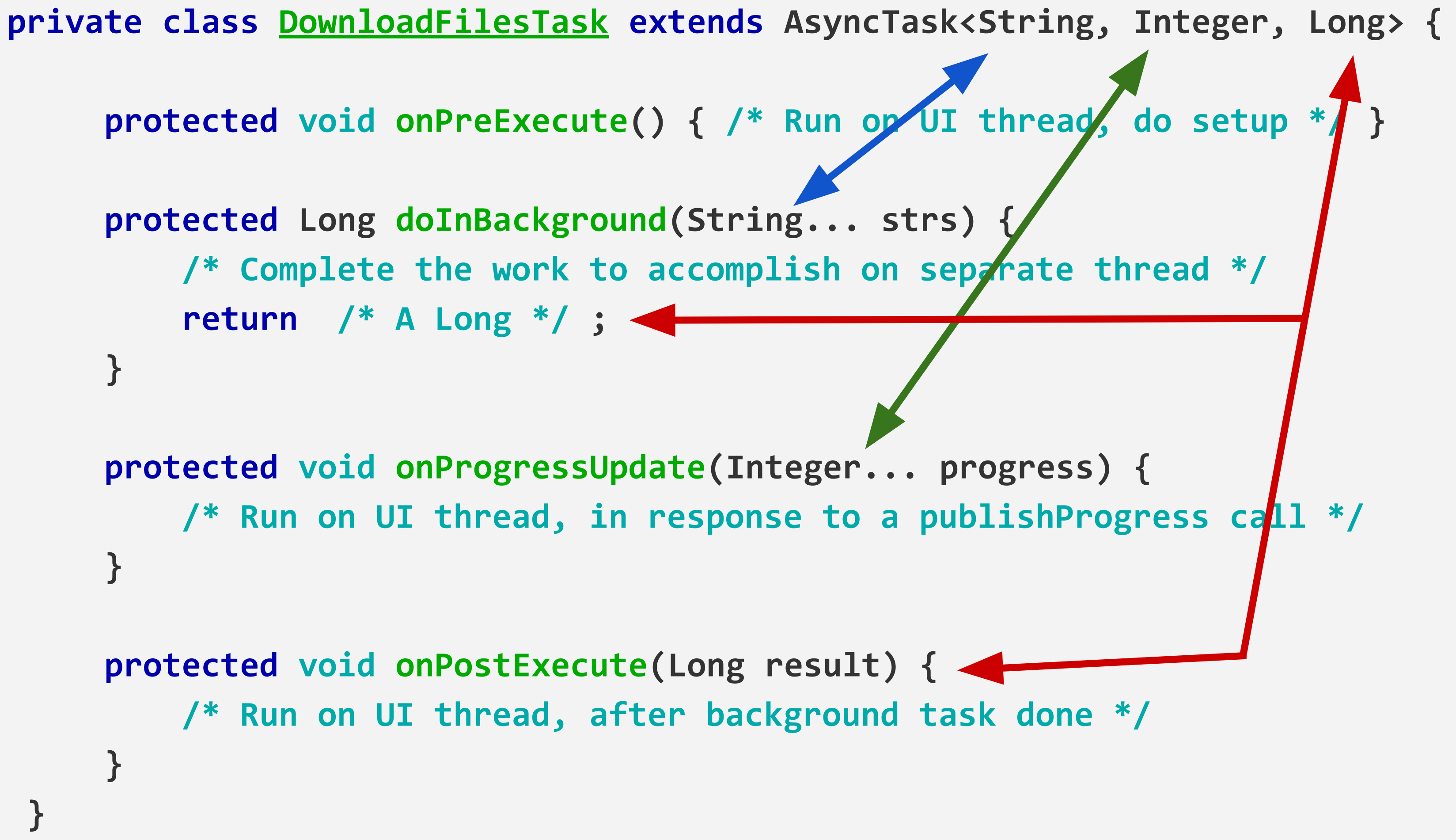
```
private class DownloadFilesTask extends AsyncTask<String, Integer, Long> {

    protected void onPreExecute() { /* Run on UI thread, do setup */ }

    protected Long doInBackground(String... strs) {
        /* Complete the work to accomplish on separate thread */
        return  /* A Long */ ;
    }

    protected void onProgressUpdate(Integer... progress) {
        /* Run on UI thread, in response to a publishProgress call */
    }

    protected void onPostExecute(Long result) {
        /* Run on UI thread, after background task done */
    }
}
```



```
private class DownloadFilesTask extends AsyncTask<String, Integer, Long> {

    protected void onPreExecute() { /* Run on UI thread, do setup */ }

    protected Long doInBackground(String... strs) {
        /* Complete the work to accomplish on separate thread */
        return  /* A Long */ ;
    }

    protected void onProgressUpdate(Integer... progress) {
        /* Run on UI thread, in response to a publishProgress call */
    }

    protected void onPostExecute(Long result) {
        /* Run on UI thread, after background task done */
    }
}
```

Prime application

- Change Prime app to use AsyncTask instead of PrimeRunnable
- Should function identically
- How much code is required?

```
private class PrimeTask extends AsyncTask
        <String, Integer, Long> {

     protected void onPreExecute() { /*Run on UI thread,setup*/ }

     protected Long doInBackground(String... strs) {
         /* Complete the work to accomplish on separate thread */
         return  /* A Long */ ;
     }

     protected void onPostExecute(Long result) {
         /* Run on UI thread, after background task done */
     }
 }
```

ICA

Replace ImageRunnable

- Change Prime app to use AsyncTask instead of **ImageRunnable**
- Should function identically
- How much code is required?

```
private class ImageAsyncTask extends AsyncTask
        <String, Integer, Bitmap> {

     protected Bitmap doInBackground(String... strs) {
         /* Complete the work to accomplish on separate thread */
         return  /* A Bitmap */ ;
     }

     protected void onPostExecute(Bitmap result) {
         /* Run on UI thread, after background task done */
     }
 }
```

MediaPlayer

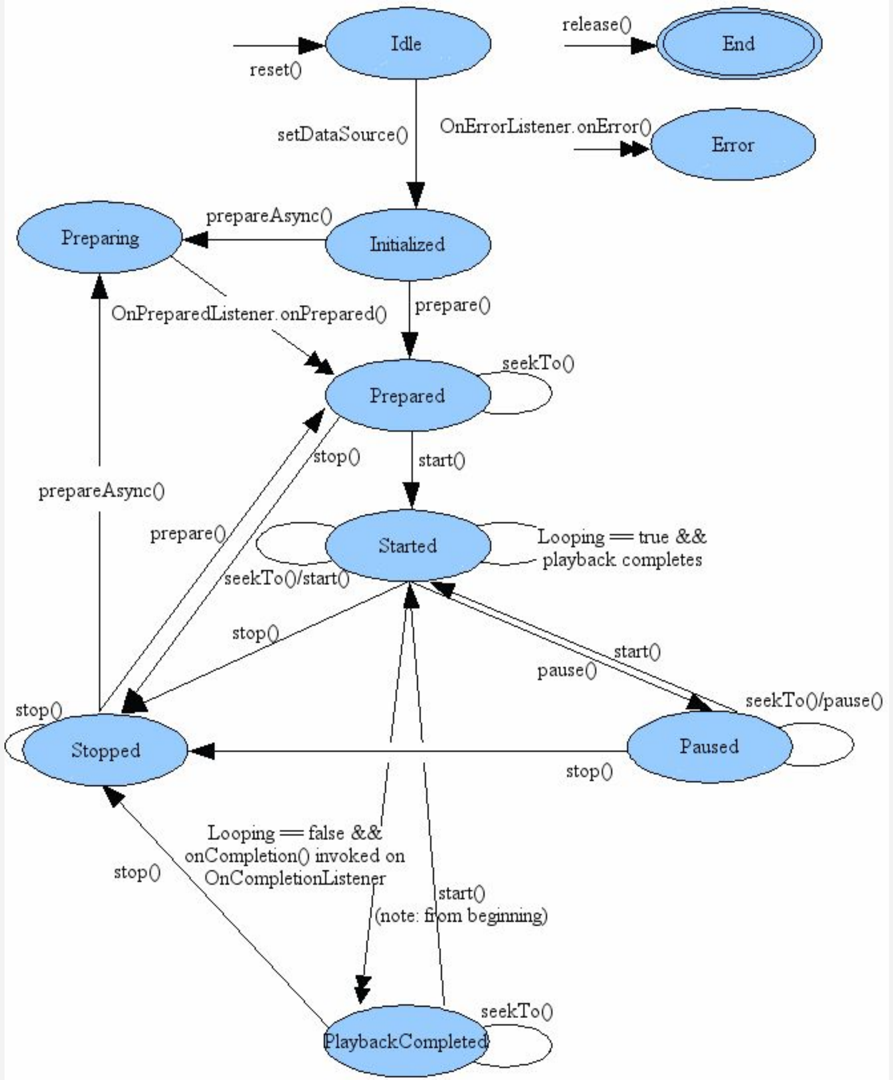
```
MediaPlayer player = MediaPlayer.create(
        context, R.raw.sound_file);

player.setLooping(false);          // don’t loop
player.setVolume(volume, volume);  // floats [0.0, 1.0]

// . . . configure  . . .

player.start();
```

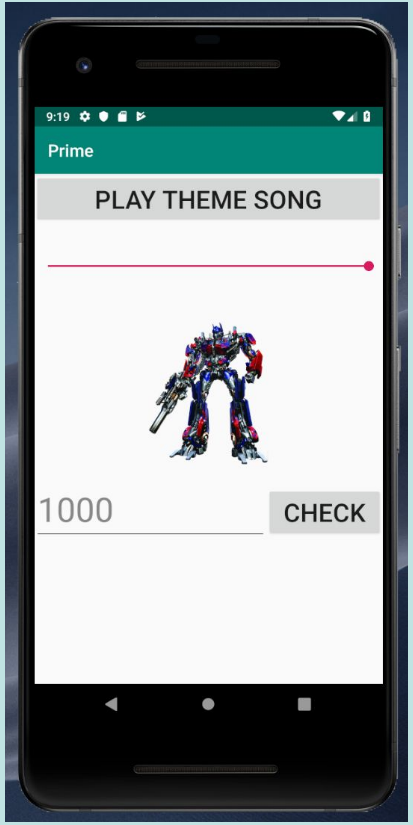
MediaPlayer State Diagram



Make a Song play when button pressed

- Start playing the sound when the app begins, or on button click
- Notice the **SeekBar**, Put it in XML
- Set it to update volume when seekbar changes

```
SeekBar volumeSeekbar =
       findViewById(R.id.volume_control);
volumeSeekbar.setOnSeekBarChangeListener(
       new VolumeListener());
```


9:19
Prime
PLAY THEME SONG
1000
CHECK