CIS472/594: Android Programming

Homework 6 – Downloader and MediaPlayer Service

Due Date: April 13, Thursday, at 11:59pm

SUBMISSION INSTRUCTIONS

1. Please submit your whole project as a .zip file on BlackBoard.
2. Demo to me after class or during my office hours.

Note: **I will give you a rubric which specifies the spec that works and not. It also needs to tell me what extra credits you implemented. Please submit the rubric.doc file along with your .zip file.**

For this homework you will develop an app that will download an MP3 file from a website provided by me and be able to play the song. Be sure to bring it all together with a clean and attractive UI.

REQUIREMENTS

- Version – your app should work for Android 4.0.3 or above (Minimum API 15)

1. Create an Activity that will be your MediaPlayer:
    - **Buttons**
        - Play Button
            - disabled when music is playing
            - enabled when music is stopped or paused
        - Stop Button
            - disabled when music is stopped
            - enabled when music is paused or playing
            - Note: you should NOT call stop() on the MediaPlayer, instead, pause() the MediaPlayer and seek to the beginning of the song.
        - Pause Button
            - disabled when music is paused or stopped
            - enabled when music is playing
    - **TextViews**
        - Song Title (TextView)
        - Artist Name (TextView)
        - Look at DownloadManager.zip to see how to get the filename by using the URL for the MP3
        - Parse the filename to get the Song Title and Artist Name

- **ImageViews**
    - Album Cover Image (ImageView):
        - if a song has not yet been downloaded, then use a default image (a good looking one) that you either need to create yourself or find somewhere else
        - if the song has been downloaded, then use this image that I have provided
        - you can place these in res/drawable/ directory (or res/drawable/xhdpi, or res/drawable/hdpi)
        - we haven't used ImageViews in class, but you should be able to figure it out within a few minutes
- **Action Bar Items**
    - a Download Button that will download this MP3
    - an Exit button that will
        - exit the app
        - stop the MediaPlayer service
        - don't stop the Download Service

if the user exits/leaves the Activity by pressing the home key or back key, the music should still be playing. Then if the user returns to the Activity, the Play/Stop Buttons should still work as expected.

2. Create a Service to download the MP3 provided

a. This **MUST** extend IntentService

b. MP3 should be downloaded when the Download option is clicked

c. you may use either DownloadManager discussed in class, which can also be found in the textbooks

d. you can hardcode the link to the MP3

3. Create another Service to handle playback of your newly downloaded song

a. This **should** extend Service

b. This **MUST** run the MediaPlayer itself inside of a Service

    i. The buttons in your Activity should make requests to the Service to play/stop/pause/etc

c. the state of the MediaPlayer should not change when the user presses the back key or home key

    i. that is, if the MediaPlayer is playing music, leaving the Activity should not stop the music

4. Manage Notifications for the following events

   a. download of the MP3 has started
   b. download of the MP3 has completed
      i. when the user clicks on this notification
         1. clear the notification
         2. launch your MediaPlayer Activity (hint: PendingIntent)
   c. the song is playing
      i. you shouldn't allow this notification to be cleared
      ii. the notification should go away when playback is stopped or paused
      iii. when the user clicks on this notification
         1. launch your MediaPlayer Activity

THINGS TO NOTE

- I don't want to take the joy of figuring things out away from you, but if you can't figure out
  - Ask me or your classmates
- If anything is unclear, please ask me. I'm very willing to help
  - Your Activity should not actually play the music itself, that's what your Service is for. Clicking on the playback Buttons from within your Activity should make calls to the Service, and the Service should take the appropriate action. This prevents playback from being stopped when your Activity loses focus
  - Do not place the MP3 inside of your res/raw directory. You're supposed to download it, then get the URI after the download, and use the URI to launch the MediaPlayer

Resources

- Album Cover
  - Download from BB
- MP3 file:
  - http://www.primetechconsult.com/CIS472/secretsong_mario.mp3

Extra credits (10 points)

1. Add a SeekBar to your Activity

   a. the SeekBar should be synced with the song (use an AsyncTask)
   b. when the user changes the progress on the SeekBar, the song should seek to the correct position
   c. the SeekBar progress should balance with the length of the song
   d. e.g. if a song is 120 seconds long
      i. moving the progress of the SeekBar to the middle of the SeekBar should seek to (or at least close to) the 60 second mark of the song
   e. stopping the song should send the ProgressBar back to 0

2. Add a Timer to your Activity using TextViews

a. the Timer should be synced with the song (use AsyncTask)
b. e.g., 90 seconds through the song, the TextViews should represent **1:30**
c. When the stop Button is pressed, the timer should go back to **0:00**
d. of course, this one is easier than adding a SeekBar to your Activity

Please see one example on next page!

EXAMPLE

Figure 1 - MediaPlayer Activity with Stop button disabled.

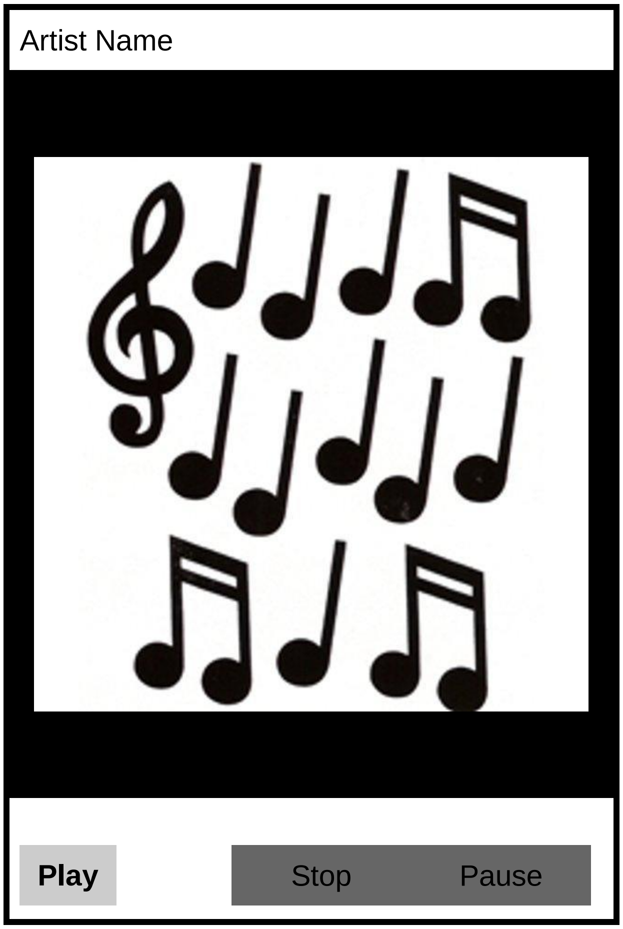
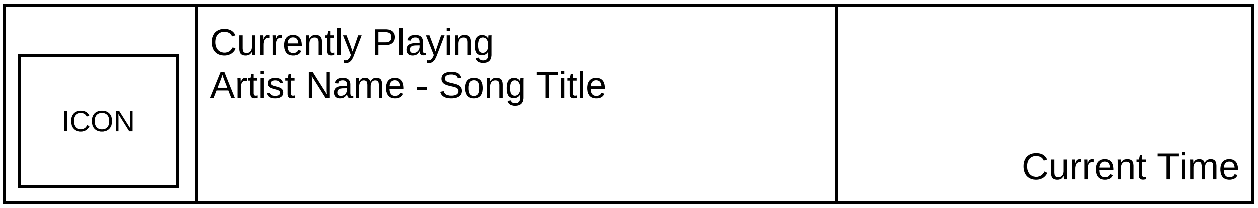


Figure 2 - Notification


ICON
Currently Playing
Artist Name - Song Title
Current Time