AudioManager audioManager = (AudioManager) getSystemService(Context.***AUDIO\_SERVICE***);

audioManager.requestAudioFocus(**this**, AudioManager.***STREAM\_MUSIC***, AudioManager.***AUDIOFOCUS\_GAIN***);

/\*\*

\* ---------------音频焦点处理相关的方法---------------

\*\*/

@Override

public void onAudioFocusChange(int focusChange) {

switch (focusChange) {

case AudioManager.AUDIOFOCUS\_GAIN://你已经得到了音频焦点。

System.out.println("-------------AUDIOFOCUS\_GAIN---------------");

// resume playback

if(isPlaying){ //

mPlayer.start();

mPlayer.setVolume(1.0f, 1.0f);

}

break;

case AudioManager.AUDIOFOCUS\_LOSS://你已经失去了音频焦点很长时间了。你必须停止所有的音频播放

System.out.println("-------------AUDIOFOCUS\_LOSS---------------");

// Lost focus for an unbounded amount of time: stop playback and release media player

if (mPlayer.isPlaying())

mPlayer.stop();

mPlayer.release();

mPlayer = null;

break;

case AudioManager.AUDIOFOCUS\_LOSS\_TRANSIENT://你暂时失去了音频焦点

System.out.println("-------------AUDIOFOCUS\_LOSS\_TRANSIENT---------------");

// Lost focus for a short time, but we have to stop

// playback. We don't release the media player because playback

// is likely to resume

if (mPlayer.isPlaying())

mPlayer.pause();

break;

case AudioManager.AUDIOFOCUS\_LOSS\_TRANSIENT\_CAN\_DUCK://你暂时失去了音频焦点，但你可以小声地继续播放音频（低音量）而不是完全扼杀音频。

System.out.println("-------------AUDIOFOCUS\_LOSS\_TRANSIENT\_CAN\_DUCK---------------");

// Lost focus for a short time, but it's ok to keep playing

// at an attenuated level

if (mPlayer.isPlaying())

mPlayer.setVolume(0.1f, 0.1f);

break;

}

}