Audio Session

사운드 콘텐츠 응용_4주차

Dept. of Multimedia Science, Sookmyung Women's University.
Prof. JongWoo Lee

Index

Audio Session Basics

Why Does iOS Need to Manage Audio?

지하철에서

아이폰을 unlock하고
아이팟을 틀었더니
내장스피커로 노래가 흘러나온다.

지하철 옆사람이
얼굴을 찡그린다.

바로 헤드셋을 꼽았더니
헤드셋으로 음악이 흘러
나온다.

음악을 듣다보니 소리가갑
자기 슬그머니 조용해지더
니 알람이 울리고, 울린 후
다시 노래가 시작된다.

- 이러한 일들이 오디오 세팅을 다시 설정하지 않아도 이루어짐
- 오디오 세션 객체를 통해 어플리케이션이 간단해지는 것이 가능

What Is an Audio Session?

- Audio Session
  - 오디오 행위 구성을 위한 어플리케이션과 iPhone OS 사이의 중개자
  - 어플리케이션의 오디오 의도(audio intention)를 표현하기 위해 사용
    - ✓ 오디오가 벨/무음 스위치에 의해 무음이 되어야 하는지
    - ✓ 오디오가 스크린 잠금의 경우 멈추어야 하는지
    - ✓ iPod과 같은 다른 오디오가 당신의 오디오가 시작될 때 계속해서 재생이 되거나 무음이 되어야 하는지
  - 어플리케이션이 구동되면 자동으로 하나 씩의 audio session을 얻음
  - Audio Session의 구성은 어플이 구동되는 동안 모든 audio 활동에 영향을 미침

개발하고자 하는 어플의 사운드를 재생할 때
어플의 audio session은
활성화(activate)되어야 하고,

전화나 알람이 울릴 때,
개발하고자 하는 어플의 audio session은
비활성화(deactivate)되어야 함

interruption: 전화가 걸려오거나, 알람이 울릴 때와 같이 오디오 세션이 비활성화되는 것

Audio Session API는 interruption을 알아차리고 반응하는 방법을 제공

What Is an Audio Session Category?

- Audio Session Category
  - 어플리케이션의 오디오 행위들을 구별하기 위한 일종의 키
  - iOS에는 6개의 카테고리가 존재

If yes,
사용자가 벨/무음 스위치를 무음으로 설정하거나, 스크린이 잠겼을 때 오디오는 음소거 됨

If yes,
다른 어플에서의 오디오가 현재 구동하려는 어플의 사운드와 함께 재생

If yes,
오디오 input(녹음)과 output(재생)이 허용

Category identifiers*	Silenced by the Ring/Silent switch and by screen locking	Allows audio from other applications	Allows audio input (recording) and output (playback)
AVAudioSessionCategoryAmbient kAudioSessionCategory_AmbientSound	Yes	Yes	Output only
AVAudioSessionCategorySoloAmbient kAudioSessionCategory_SoloAmbientSound	Yes	No	Output only
AVAudioSessionCategoryPlayback kAudioSessionCategory_MediaPlayback	No	No by default; yes by using override switch	Output only
AVAudioSessionCategoryRecord kAudioSessionCategory_RecordAudio	No (recording continues with the screen locked)	No	Input only
AVAudioSessionCategoryPlayAndRecord kAudioSessionCategory_PlayAndRecord	No	No by default; yes by using override switch	Input and output
AVAudioSessionCategoryAudioProcessing kAudioSessionCategory_AudioProcessing	-	No	No input and no output

← Default

Audio Session Default Behavior

- Default Behavior
  - 사용자가 벨/무음 스위치를 무음에 가져가면 오디오는 무음이 된다.
  - 사용자가 sleep/wake 버튼을 눌러 스크린을 잠그거나 자동 잠금 시간이 지나면 오디오는 무음이 된다.
  - 오디오가 시작되었을 때, 기기의 다른 오디오(이미 재생되고 있는 iPod 오디오)는 무음이 된다.
  - 재생은 사용할 수 있지만, 녹음은 사용할 수 없다.

- Default Audio Session으로 부족한 이유
  - 사용자가 오디오북 어플리케이션을 실행 중이라고 할 때,
  - 몇분이 지나면 auto-lock시간이 되어 스크린은 잠기고 오디오는 무음이 됨
  - 스크린이 잠겨도 계속해서 재생이 되게 하려면 AVAudioSessionCategoryPlayback 카테고리를 사용

How the System Resolves Competing Audio Demands

- 오디오 세션의 흐름

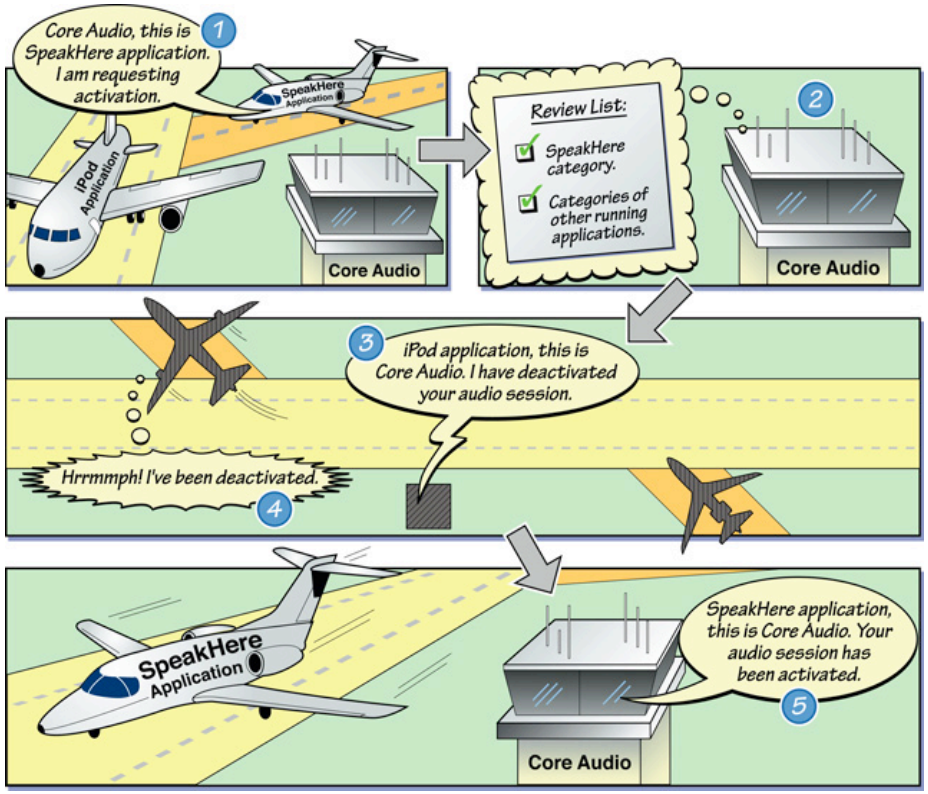
상황: iPod이 이미 재생되고 있는 동안 다른 어플리케이션이 오디오를 사용하기 원할 경우

1. SpeakHere App이 오디오 세션의 활성화를 요청(App이 구동되거나 App에서 플레이 버튼을 사용자가 탭했을 때)

2. 오디오 세션에 지정된 카테고리를 가지고 Core Audio가 활성화 요청을 고려(SpeakHere App은 다른 오디오를 무음으로 만드는 카테고리를 사용)

3, 4. Core Audio가 iPod App의 오디오 세션을 비활성화 시키고, 오디오 재생을 멈춤

5. Core Audio가 SpeakHere App의 오디오 세션을 활성화하고 재생을 시작



The Two Audio Session APIs

- AVAudioSession class (AV Foundation framework사용)
  - 오디오 세션의 Core Set에 접근을 제공하는 Objective-C API
  - 세션은 암시적으로 초기화(C API와 같이 분리되어 초기화가 실행되지 않음)
  - Sample rate나 channel count와 같이 하드웨어의 configuration을 변경하고, 오디오 interruption을 다루기 위해 delegate method를 사용(재생, 녹음을 하는 중에도 어떤 오디오 기술을 사용하든 이 delegate method를 사용할 수 있음)
- Audio Session Services
  - 오디오 세션의 모든 basic, advanced 기능의 접근을 제공하는 full-featured C API
  - interruption을 달기 위한 callback 메커니즘을 제공

이 두가지 API는 mix & match하게 호출 가능하며
서로 완벽하게 호환됨

Developing with the Audio Session APIs

- 개발하고자 하는 어플리케이션에 오디오 세션을 추가하여도 시뮬레이터에서는 오디오 세션의 동작을 살펴볼 수 없음

- 시뮬레이터에서 실행할 수 없는 동작
  - interruption을 호출하는 것
  - 벨/무음 스위치 설정을 변경하는 것
  - 스크린 잠금
  - 헤드셋 플러그/언플러그
  - 오디오 세션 카테고리 행위를 테스트하는 것
  - audio mixing 행위를 테스트하는 것

Configuring the Audio Session

Initializing the Audio Session

- 어플리케이션 구동 시, 시스템은 audio session object를 제공
- 세션이 동작하기 전에 반드시 초기화
- AudioSession class 사용시

```
// implicitly initializes the audio session
AVAudioSession session = [AVAudioSession sharedInstance];
```

- Audio Session Service 사용시
  - AudioSessionInitialize function으로 명시적 초기화를 사용하는 오디오 세션 코드를 추가
  - 일반적으로 main controller class에서 명시적인 오디오 세션 초기화를 수행

```
AudioSessionInitialize (
            NULL,                               // inRunLoop NULL = default run loop
            NULL,                               // inRunLoopMode NULL = default run loop mode
            interruptionListenerCallback,       // inInterruptionListener
            userData                            // inClientData
);
```

Activating and Deactivating the Audio

- 시스템이 어플리케이션이 실행될 때 오디오 세션을 활성화하지만 Apple은 세션을 명시적으로 활성화할것을 권고
- viewDidLoad로 성공적으로 활성화 되었는지 아닌지 테스트
- AudioSession class 사용시

```
NSError *activationError = nil;
[[AVAudioSession sharedInstance] setActive: YES error: &activationError];
```

  - Deactivate를 위해서는 setActive를 No로 설정

- Audio Session Service 사용시

```
OSStatus activationResult = NULL;
result = AudioSessionSetActive (true);
```

  - Deactivate를 위해서는 AudioSessionSetActive를 false로 설정

Choosing the Best Category

- To pick the best category

어플리케이션에서 오디오가 꼭 필요한가?

If essential,
벨/무음 스위치가 무음이어도 재생을 지원하는 카테고리 선택
AVAudioSessionCategoryPlayback

If peripheral,
벨/무음 스위치가 무음이면 오디오도 무음이 되는 카테고리 선택
AVAudioSessionCategorySoloAmbient

구동하고자 하는 어플리케이션의 오디오와
본래 구동되고 있는 오디오가
같이 재생되어야 하는가?

If 게임 app 실행 시,
iPod의 오디오가 이미 재생되고 있어도 음악을 들으며 게임을
할 수 있게 기존 오디오가 계속 재생하게 하는 카테고리 선택
AVAudioSessionCategoryAmbient

Setting the Category

- AudioSession class 사용시

```
NSError *setCategoryError = nil;
[[AVAudioSession sharedInstance]
            setCategory: AVAudioSessionCategoryAmbient error: &setCategoryError]; //카테고리 지정

if (setCategoryError) { /* handle the error condition */ }
```

- Audio Session Service 사용시

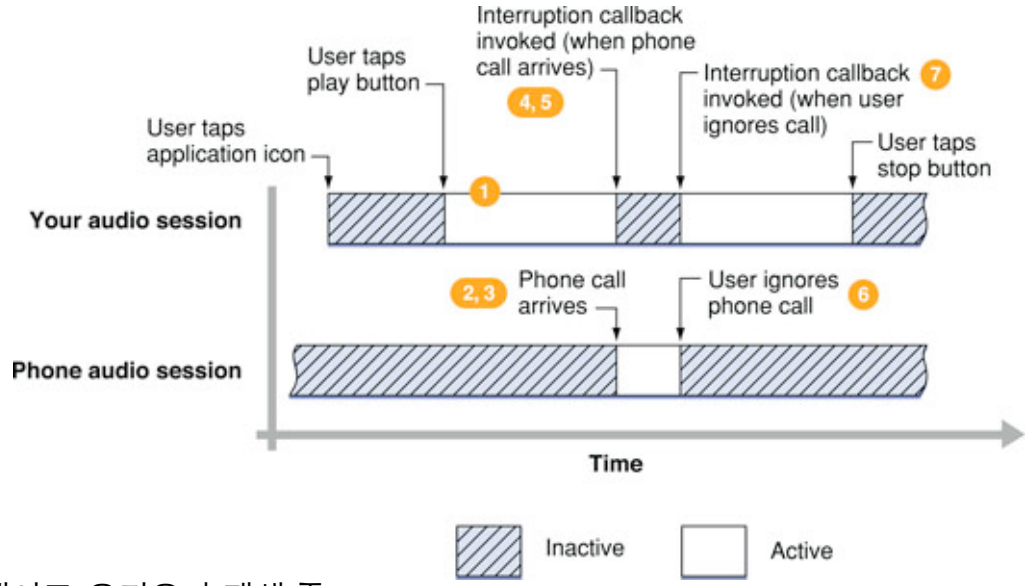
```
UInt32 sessionCategory = kAudioSessionCategory_AmbientSound;          // 1

AudioSessionSetProperty (
            kAudioSessionProperty_AudioCategory,                      // 2
            sizeof (sessionCategory),                                 // 3
            &sessionCategory                                          // 4
);
```

1. 새로운 타입의 UInt32로 정의하고 오디오 세션에 적용하길 원하는 식별자로 초기화
2. 원하는 오디오 세션 속성값을 위한 식별자
3. 속성값의 크기
4. 카테고리

Handling the Audio Session

The Interruption Life Cycle

재생 어플리케이션을 위한 오디오 세션 interruption의 시작 전, 중간, 후의 이벤트 단계를 설명



1. 재생 어플리케이션이 활성화 상태이고, 오디오가 재생 중
2. 전화가 걸려온다. 시스템은 폰 어플리케이션의 오디오 세션을 활성화
3. 시스템은 오디오 세션을 비활성화 시킴, 이 시점에서 오디오 어플리케이션의 재생은 정지
4. 시스템은 오디오 어플리케이션이 비활성화할 것을 지시하면서 interruption listener callback function(or interruption started delegate method)을 호출
5. callback(or delegate method)는 적절한 액션을 취함
   Ex) 사용자 인터페이스를 업데이트 하거나 멈췄던 지점에서 다시 재생하기 위해 필요한 정보를 저장
6. 만일 사용자가 interruption을 해제할 때(전화가 걸려오는 것을 무시하는 것을 선택하는 것)
   시스템은 interruption이 종료될 것을 지시하면서 callback 또는 delegate method를 호출
7. callback 또는 delegate method가 interruption이 종료되면서 적절한 액션을 취함
   Ex) 사용자 인터페이스를 업데이트하고, 오디오 세션을 재활성화하고, 재생을 다시 시작
8. (위의 그림에는 나와있지 않은)만일 6번 단계에서 interruption이 해제되는 것 대신 사용자가 전화를 받았을 경우 어플리케이션은 중지

Audio Interruption Handling Techniques

- interruption에 반응하기 위한 두가지 접근
  - Objective-C interruption delegate method를 구현(simpler & fit better)
  - C-based interruption callback function을 작성(명시적으로 오디오 세션 초기화를 호출해야 함)

- 오디오 세션 interruption 동안 발생하는 일들

After interruption starts	• Check whether resumption of audio process is supported • Save state and context • Update user interface
After interruption ends	• Restore state and context • Reactivate audio session • Update user interface

Handling Interruption Using Delegate Method(1/3)

- AVAudioSession의 AVAudioSessionDelegate 사용하는 방법
  - *beginInterruption* (다음 장에 샘플 코드 참고)
    - ✓ 오디오 세션이 중단된 후에 호출
    - ✓ 현재 어플리케이션이 막 종료되려고 할 것을 가정하여 이 method를 구현(사용자가 걸려오는 전화를 받았을 때와 같이)

      ex) 어플리케이션의 다음 구동 시, 사용자가 interruption의 위치에서부터 재생이 시작되는 것을 원할 경우, 현재 file frame number와 file identifier를 저장
  - *endInterruption*
    - ✓ 오디오 세션 interruption이 종료된 후에 호출. 반드시 오디오 세션을 명시적으로 재활성화

      (AVAudioPlayer or AVAudioRecorder 객체는 자동으로 오디오 세션이 재활성화)

- AVAudioPlayer or AVAudioRecorder class 사용하는 방법
  - *audioPlayerBeginInterruption* 오디오 세션이 재생되는 동안 중단되었을 때 호출
  - *audioPlayerEndInterruption* 재생 interruption이 종료되었을 때 호출
  - *audioRecorderBeginInterruption* 오디오 세션이 녹음 도중 중단되었을 때 호출
  - *audioRecorderEndInterruption* 녹음 interruption이 종료되었을 때 호출
  - 21페이지 샘플코드 참고

Handling Interruption Using Delegate Method(2/3)

- Using the AVAudioSessionDelegate Protocol 사용 예
  - *beginInterruption*
    - ✓ 오디오 세션이 중단된 후에 호출
  - *endInterruption*
    - ✓ 오디오 세션 interruption이 종료된 후에 호출

```
-(void) beginInterruption {                    //오디오 세션이 interrupted 되었을 때
    if (playing) {                             //재생중이라면
        playing = NO;                          // 재생 정지
        interruptedWhilePlaying = YES;
        [self updateUserInterface];
    }
}

 NSError *activationError = nil;
-(void) endInterruption {                      //interruption이 종료되었을 때
    if (interruptedWhilePlaying) {
        [[AVAudioSession sharedInstance] setActive: YES error: &activationError];
        [player play];
        playing = YES;
        interruptedWhilePlaying = NO;
        [self updateUserInterface];
    }
}
```

Handling Interruption Using Delegate Method(3/3)

- Using the AVAudioPlayer Class 사용 예
  - *audioPlayerBeginInterruption* 오디오 세션이 재생되는 동안 중단되었을 때 호출
  - *audioPlayerEndInterruption* 재생 interruption이 종료되었을 때 호출
  - *audioRecorderBeginInterruption* 오디오 세션이 녹음 도중 중단되었을 때 호출
  - *audioRecorderEndInterruption* 녹음 interruption이 종료되었을 때 호출

```
-(void) audioPlayerBeginInterruption: (AVAudioPlayer *) player {
      if (playing) {
          playing = NO;
          interruptedOnPlayback = YES;
          [self updateUserInterface];
      }
}
```

```
- (void) audioPlayerEndInterruption: (AVAudioPlayer *) player {
       if (interruptedOnPlayback) {
           [player prepareToPlay];
           [player play];
           playing = YES;
           interruptedOnPlayback = NO;
        }
}
```

Handling Interruption Using a Callback Function(1/3)

- Audio Session Service의 *AudioSessionInterruptionListener* callback prototype 사용하는 방법

```
typedef void (*AudioSessionInterruptionListener) (
            void *inClientData,
            UInt32 inInterruptionState
);
```

- Two Interruption State
  - *kAudioSessionBeginInterruption*
    - ✓ 오디오 세션은 지금 막 중단되었고, 비활성화 됨
    - ✓ 어플리케이션이 종료되기 전, 필요한 행동을 수행하기 위한 *interruption callback*을 사용
  - *kAudioSessionEndInterruption*
    - ✓ Interruption이 종료 됨

      ex) 사용자가 걸려오는 전화를 무시할 것을 선택했을 때, callback은 재생을 재시작, 오디오 세션을 재활성화할 것을 전달
- Callback 사용 시, *interruption*을 지원하는 *3*가지 방법
  - *Define interruption method (*다음장에 계속*)*
  - *Define a callback function (*다음장에 계속*)*
  - *Register the callback function with audio session (12*페이지의 두번째 코드 참고*)*

Handling Interruption Using a Callback Function(2/3)

- Defining Interruption Methods
  - Interruption-begin, interruption-end시 적절한 액션을 취하게 하는 method 구현

```
-(void) resumePlayback {

        UInt32 sessionCategory = kAudioSessionCategory_MediaPlayback;       // 1
        AudioSessionSetProperty (                                           // 2
            kAudioSessionProperty_AudioCategory,                            // 3
            sizeof (sessionCategory),                                       // 4
            &sessionCategory                                                // 5
        );
        AudioSessionSetActive (true);                                       // 6

        [self.audioPlayer resume];                                          // 7
}
```

1. 세션 카테고리 정의, media playback 카테고리 식별자로 초기화 (*kAudioSessionCategory_MediaPlayback* : 오디오 세션이 활성화되어있을 때, 폰의 다른 오디오는 무음으로 설정)
2. AudioSessionSetProperty 함수는 지정된 속성값을 assign
3. Audio session property
4. 속성의 사이즈
5. 변수의 주소
6. 오디오 세션 활성화
7. audioPlayer 객체의 resume 함수 호출

Handling Interruption Using a Callback Function(3/3)

- Defining an Interruption Listener Callback Function
  - 2개의 interruption state에 반응하는 function정의

```
void interruptionListenerCallback (
      void *inUserData,                                                    // 1
      UInt32 interruptionState                                             // 2
) {
      AudioViewController *controller = (AudioViewController *) inUserData;  // 3
      if (interruptionState == kAudioSessionBeginInterruption) {           // 4
            if (controller.audioRecorder) {
                  [controller recordOrStop: (id) controller];            // 5
            } else if (controller.audioPlayer) {
                  [controller pausePlayback];                            // 6
                  controller.interruptedOnPlayback = YES;                // 7
            }
      } else if ((interruptionState == kAudioSessionEndInterruption) &&
                                                controller.interruptedOnPlayback) { // 8
            [controller resumePlayback]; controller.interruptedOnPlayback = NO;
      }
}
```

1. 오디오 세션을 초기화했을 때 제공되는 데이터의 포인터
2. Interruption state
3. AudioViewController 객체 초기화
4. If true, 오디오 세션 interruption이 막 시작되었고, 오디오 재생이나 녹음이 막 정지됨
5. If 계속 녹음 중, recoding buffer를 flush하고, 정지된 상태를 보여주는 사용자 인터페이스를 update
6. If 계속 재생 중, 재생 상태를 저장하고, 정지된 상태를 보여주는 사용자 인터페이스를 update
7. 재생이 일시적으로 중단된 곳을 지시하는 flag 설정
8. If interruption이 제거 되었다면, 어플리케이션의 오디오 다시 재생

OpenAL and Audio Interruption

- AVAudioSession delegate method 또는 interruption listener callback function으로 구현
- Interruption 코드는 반드시 OpenAL context에서 다루어야 함

```
void openALInterruptionListener (
        void *inClientData,
        UInt32 inInterruptionState
) {
        if (inInterruptionState == kAudioSessionBeginInterruption) {
                alcMakeContextCurrent (NULL);                   //disable context
        } else if (inInterruptionState == kAudioSessionEndInterruption) {
                alcMakeContextCurrent (myContext);              //set active context
        }
        // other interruption-listener handling code
}
```