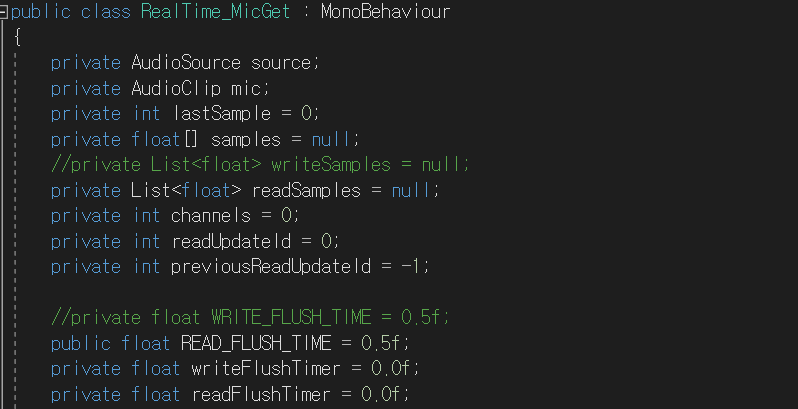
**Capturing User’s Speech (Utterance) via Microphone in Unity : Code Assignment 1**

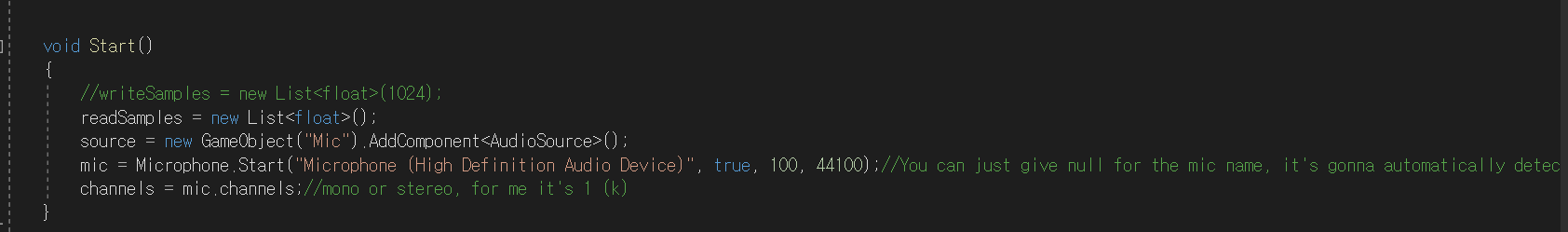
메타버스엔터테인먼트 전공 V2022123 이혜민

**Script 설명**

****

역할: 스크립트에서 사용할 변수들을 선언한다.

AudioSource는 source라고 선언하고, AudioClip은 mic로 선언한다.



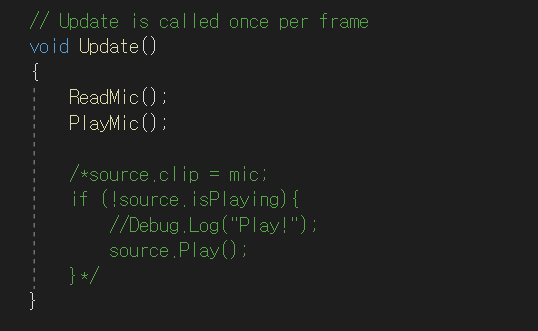
역할: 스크립트를 시작할 때 특별한 조건 없이 start 함수를 가장 먼저 실행한다.

앞서 선언한 readSample를 리스트 형으로 선언하고,

Source를 “Mic”라는 게임오브젝트(GameObject)를 AudioSource라는 컴포넌트를 추가한 것으로 다시 정의한다.

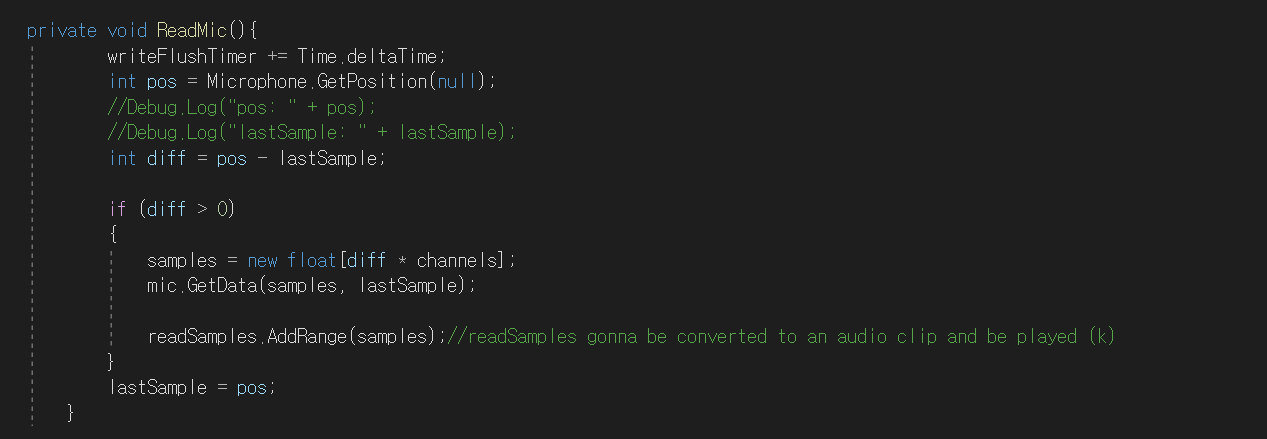
Microphone은 유니티에서 마이크 입력을 처리해주는 클래스이다. 따라서 Microphone.start를 하게 되면 파라미터로 deviceName, loop, lengthSec, frequency를 필요로 한다. 여기서는 deviceName은 ‘기기의 명칭’을 의미하고, 여기서는 “Microphone (High Definition Audio Device)”가 기기의 명칭이 된다. 해당 deviceName 파라미터는 null값으로 받을 수 있으며, null값인 경우 디폴트 마이크로 설정된다. Loop는 lengthSec(녹음에 의해 생성된 AudioClip의 길이)에 도달하게 되면, 녹음이 계속 녹음 되어야 하는지 여부를 나타내는 값으로 소스 상에는 true값으로 설정하여 계속 녹음이 되게 만들었다. 그리고 lengthSec는 녹음된 AudioClip의 길이로 기본 단위는 초인데, 소스 상에는 100초 동안 녹음되게 하였다. 그리고 frequency는 빈도를 의미하여, ‘녹음에 의해 생성된 AudioClip의 샘플 속도’를 의미한다. 여기서는 44100Hz로 설정되어 있다.

Channel은 mic의 channels로 선언하는 데, 여기서는 채널은 1개로 지정했다.



업데이트 함수는 프레임당 한번 호출되는 함수로 start 다음에 실행된다.

여기서는 ReadMic()과 PlayMic()라는 자체적으로 만든 함수들이 있는데, 실행되는 순서는 ReadMic() 🡪 PlayMic() 로 진행된다.



Time.deltaTime은 ‘전 프레임이 완료되기까지 걸린 시간’을 의미한다. PC의 성능과 무관하게 동등한 성능 조건이 되기 위해서 Time.deltaTime을 사용하며, Time.deltaTime이 증가할 때마다 writeFlushTimer에 더해준다.

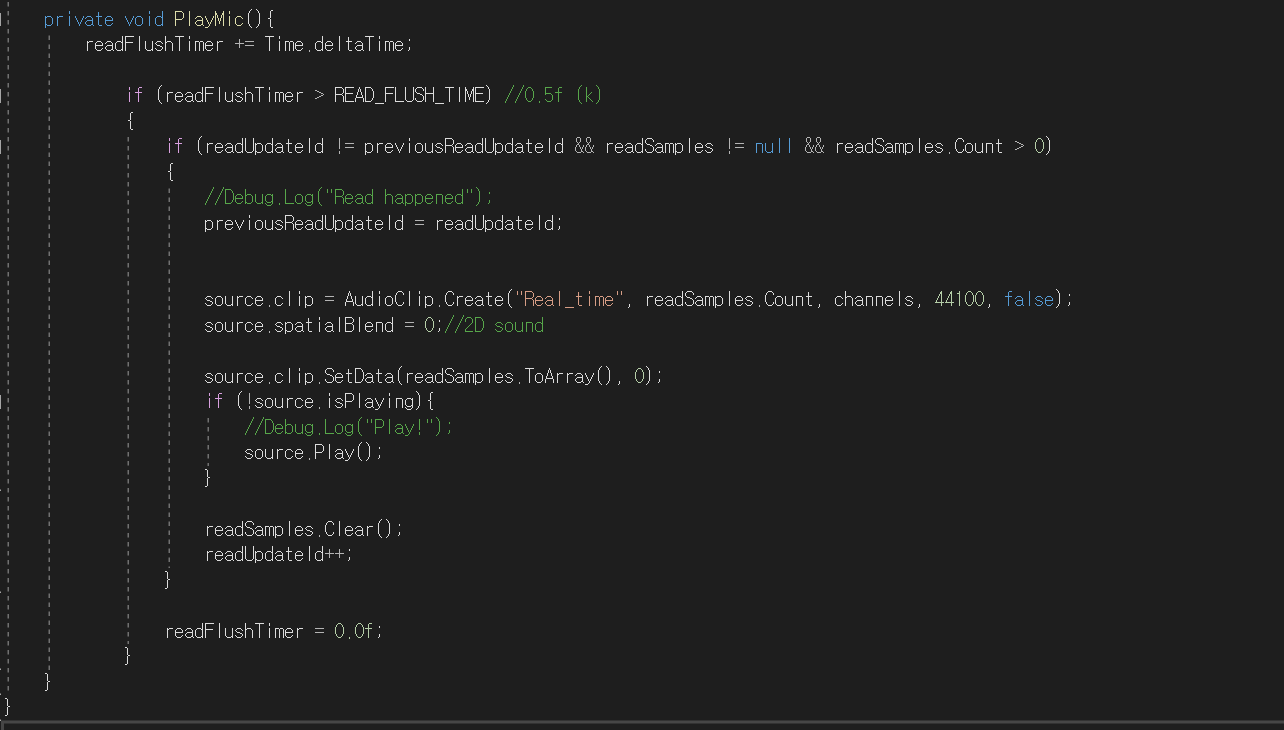
정수형 pos에다가 microphone의 GetPosition 함수값을 넣어 현재 레코딩 샘플 위치값이 들어가게 선언했다.

그리고 정수형 diff를 레코딩 샘플 위치값(pos)에서 선언부에서 선언한 lastSample(0)을 뺀 값으로 정의했다.

만약 레코딩 샘플 위차값이 존재한다면 samples는 pos에서 channels을 곱한 값으로 정의했고,

Mic의 샘플 데이터의 배열을 가져오는 함수를 호출했다. GetData의 파라미터로는 float형의 samples와 int의 lastSample이 사용되었다. 그리고 readSamples.AddRange는 readSample의 배열의 끝에 항목을 집어넣는 함수로 samples(pos \* channels)을 추가한다.

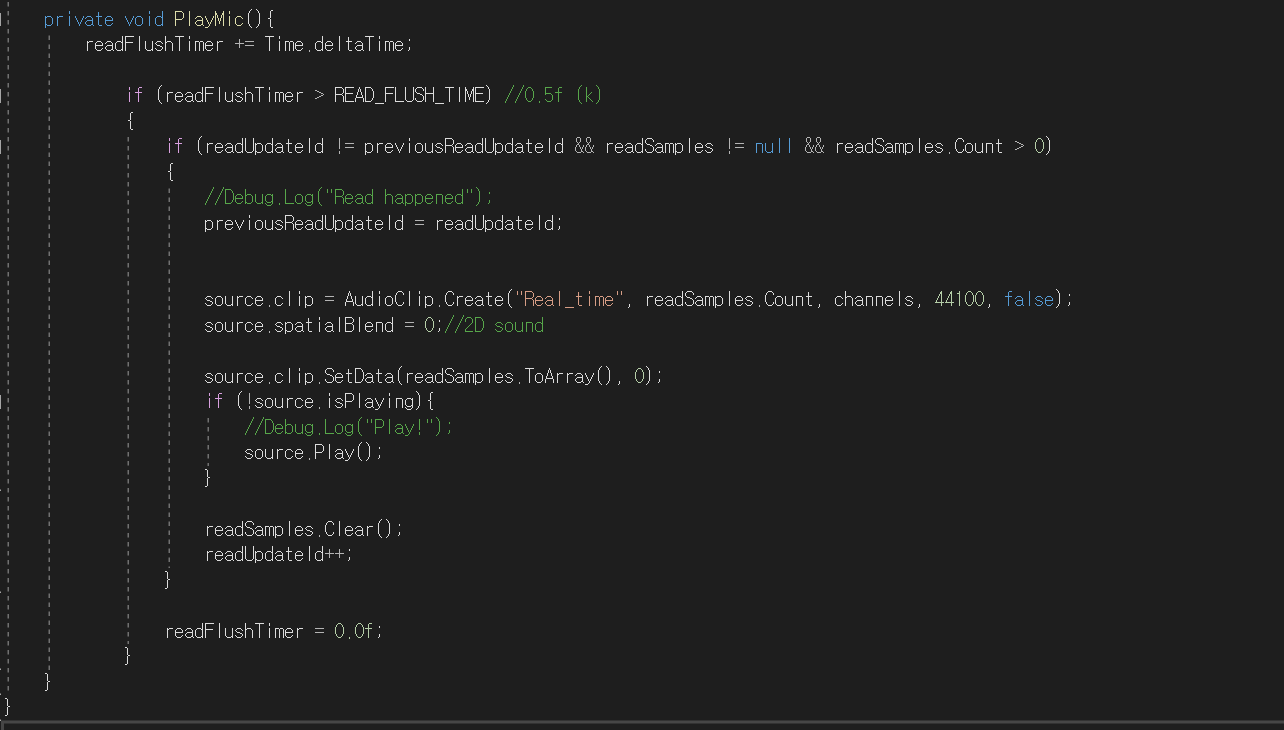
그러나 diff값이 0보다 크지 않다면(pos – lastSamples), lastSamples은 pos값과 동일하다고 정의하고 있다.



PlayMic()에선 readFlushTimer를 Time.deltaTime 누적값으로 정의했다.

만약 readFlushTimer가 앞서 선언한 READ\_FLUSH\_TIME인 0.5f보다 클 경우

if문에서 readUpdateId는 previousReadUpdateId랑 readSamples가 null이 아니어야 하고, readSamples의 개수가 0보다 클 때 -1이었던 previousReadUpdateId는 readUpdateId값으로 변경된다. 그리고 기존의 오디오소스인 source의 clip은 name, lengthSamples, channels, frequency,



stream의 파라미터를 받아 “Real\_time”이라는 명칭을 가지고, readSamples의 개수만큼의 sample 프레임 개수를 가지며, 기존의 channel만큼의 프레임당 채널 개수와 44100Hz의 샘플 주파수, 스트리밍 여부를 false로 가지는 오디오클립을 생성한다.

그리고 해당 오디오클립의 3d 공간 계산 시 영향받는 정도를 spatialBlend라 하는데, 0이라 설저하여 해당 사운드가 풀 2D임을 보여주고 있다.

그리고 source의 오디오 데이터를 설정하기 위해 SetData 함수를 쓰는데, 여기서는 readSamples의 배열과 0번째 위치에 해당 오디오 데이터를 설정한다.

이때 오디오클립의 isPlaying 값이 true라면(=AudioSource.Pause가 호출되지 않으면) 오디오 클립이 재생되게 하고, readSamples값을 초기화한 다음, readUpdateId값을 1씩 증가시킨다.

그리고 readFlushTimer의 값을 0.0f로 설정한다.