과제 1: Capturing User’s Speech (Utterance) via Microphone in Unity

## 코드 분석

**텍스트, 화면, 스크린샷이(가) 표시된 사진

자동 생성된 설명**

public class RealTime\_MicGet의 초기 변수들이 정의된 후 Start()의 코드들이 1회 실행됨. Start()에서는 이후 Update의 코드들에 사용될 오브젝트를 인스턴스화함. 예를 들어, 오디오 클립을 재생, 정지, 수정을 하는 컨트롤러 역할을 수행하는 “Mic” 오디오 소스 게임 오브젝트가 생성됨. 더불어, Microphone 클래스의 Start 메소드와 같이, 클래스의 메소드는 별도의 객체를 생성하지 않고 사용이 가능한 것을 확인할 수 있음.

**코드 정의**

**[C#] class**

클래스로 여러 객체를 만들어 낼 수 있음. 객체는 서로 독립적이며 구별 가능한 실체임. 각 객체는 상태(state)와 행동(behaviour)을 지님. class에서 상태는 field로 나타냄.

클래스명은 대문자로 시작하며 클래스의 **메소드**는 인스턴스를 안 만들고 클래스 이름만으로도 사용 가능.

예시)

텍스트이(가) 표시된 사진

자동 생성된 설명

**[Unity] MonoBehaviour**

텍스트, 시계이(가) 표시된 사진

자동 생성된 설명

출처: RealTime\_MicGet.cs

MonoBehavior는 유니티에서 생성하는 모든 스크립트가 상속받는 기본 **클래스**임. 상속이란 미리 작성된 스크립트를 필요시 통째로 불러와 사용하는 것임. 위 예시처럼 클래스 이름 옆에 ‘:’을 써주고 상속받을 스크립트 명을 붙여줌.

MonoBehaviour 스크립트에서 Unity에서 가장 빈번하게 사용되는 메소드는 Start와 Update임. **Start** 메소드에 적힌 코드들은 스크립트가 처음 시작할 때 한번 실행됨. 반면, **Update** 메소드에 적힌 코드들은 매 프레임마다 반복해서 실행됨. 실행되는 순서는 Start 그 다음 Update임.

**[C#] public, private**

Public, private은 클래스에 붙어서 접근 권한을 설정함. Public은 모든 외부 객체로부터 접근을 허용함. Private 은 해당 클래스 내에서만 사용됨. 외부 혹은 파생 클래스에서 접근이 불가함.

텍스트이(가) 표시된 사진

자동 생성된 설명

출처: RealTime\_MicGet.cs

**[Unity] Audio Source**

오디오 소스는 씬에서 오디오 클립을 재생하며 이 클립은 오디오 리스너 (Audio Listner) 또는 오디오 믹서 (Audio Mixer)를 통해 재생 가능함. 오디오 소스는 실제 사운드 파일인 오디오 클립이 할당되어야지 작동하며, 오디오 소스는 이 클립을 재생, 정지, 수정을 하는 컨트롤러 역할을 함.

**[Unity] Audio Listener**

씬에서 주어진 오디오 소스로부터의 입력을 수신하여 컴퓨터 스피커를 통해 사운드를 재생하는 역할을 수행함. 프로퍼티가 없으므로 작업에 추가되어야 하며 기본값으로 항상 메인 카메라에 추가되어 있음. 오디오 리스너를 메인 카메라 또는 플레이어를 나타내는 게임 오브젝트(Game Object) 중 하나에 추가해야 하며, 각 씬에서 제대로 작동할 수 있는 오디오 리스너는 한 개뿐임.

**[C+] float f**

Float은 숫자 뒤에 f를 붙임

**[C+] List<float>**

Float 값을 가지는 원소로 이루어진 list

텍스트이(가) 표시된 사진

자동 생성된 설명

**[C+] float[]**

Float 값을 가지는 원소로 이루어진 array

텍스트이(가) 표시된 사진

자동 생성된 설명

**텍스트이(가) 표시된 사진

자동 생성된 설명**

출처: RealTime\_MicGet.cs

**[C+] new**

New 연산자는 새로운 인스턴스를 만듦.

**[Unity] GameObject**

Unity의 GameObject 클래스는 씬 내에 존재할 수 있는 모든 요소 (눈에 보이지 않는 카메라, 빛 포함)를 나타내는 데 사용됨. GameObject는 클래스이며 객체가 아님.

**[Unity] gameObject**

오브젝트는 객체임. 컴포넌트들을 부품으로서 가지며, 게임 오브젝트에 컴포넌트를 추가함으로써 게임 오브젝트가 캐릭터가 되는지 아이템이 되는지 배경이 되는지 등 역할이 결정됨. 즉, 게임 오브젝트는 기능을 구현하는 컴포넌트들을 담는 하나의 컨테이너 역할을 수행함. 게임 오브젝트는 이름, 태그, 레이어를 가지고 있으며, 후에 원하는 오브젝트를 찾는 작업에 사용될 수도 있음.

**[Unity] Component**

모든 게임 오브젝트에 붙여진 기본 클래스임. 오브젝트에 붙일 수 있는 다양한 기능을 가진 구성요소로서, 게임 오브젝트의 역할을 결정함. 유니티에서 제공하는 기본적인 컴포넌트 외에도 개발자가 집적 생성한 컴포넌트를 게임 오브젝트에 붙일 수 있음. 변수로는 gameObject, tag, transform이 있음.

**[Unity] AddComponent**

public Component AddComponent (string className); 컴포넌트 클래스를 게임 오브젝트에 추가함.

public Component AddComponent (Type componentType);

public T AddComponent (); componentType의 타입을 가진 컴포넌트 클래스를 게임 오브젝트에 추가함.

Script 클래스의 이름을 게임 오브젝트에 추가함으로써 게임 오브젝트에 스크립트를 추가할 수 있음. 일부 컴포넌트는 다른 컴포넌트가 동일한 게임 오브젝트에 있는 것을 요구하며, 이는 자동으로 추가됨. 클래스 인스턴스를 가지고 오면 암묵적으로 인스턴스가 생성됨. **인스턴스화**는 클래스로 객체를 찍어내 실체화하는 것을 뜻함.

**[Unity] this**

텍스트이(가) 표시된 사진

자동 생성된 설명

**[Unity] Microphone**

연결된 마이크로 AudioClip으로 녹음을 하는 데에 사용하는 클래스임. 메소드로는 대표적으로 Start, End가 있고, 프로퍼티로는 devices가 있으며 devices로 사용 가능한 마이크 디바이스의 리스트를 얻을 수 있음.

텍스트이(가) 표시된 사진

자동 생성된 설명

**[Unity] Time.deltaTime**

이전 프레임에서 현재 프레임까지 걸린 시간(seconds)으로 Update method가 호출될 때마다 이전에 update method가 호출되었을 때부터 현재 update method가 호출되었을 때까지의 누적 시간을 나타냄. 초당 프레임 횟수 (FPS)를 사용하면 PC 성능에 따라 값이 상이해지기 때문에 Time.deltaTime을 이용하여 PC 성능과 무관하게 동등한 조건을 만듦.