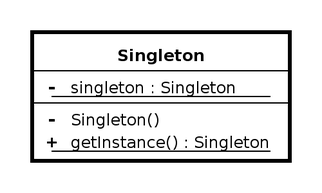
Unity3D 제네릭(Generic)을 적용한 싱글톤 매니져 만들기



제네릭 싱글톤 만들기

지난 번에 소개해 드린 싱글톤 형태로 구현을 하게되면, 매니져를 구성해서 사용할 수 있다는 장점이 있지만 문제느 매니져 클래스 마다 해당 코드를 다시 작성해야 하는 부담이 있다는 점입니다. 그 대신 아래 처럼 제네릭 매니져 클래스를 생성하고, 이를 < > 꺽쇠 괄호를 이용해서 상속하면 쉽게 사용이 가능합니다.

예제 코드를 보면서 확인해보겠습니다. 아래 스크립트 내용을 작성합니다.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | using UnityEngine;  using System.Collections;    public class Singleton<T> : MonoBehaviour where T : MonoBehaviour  {      private static T \_instance;        public static T Instance      {          get          {              if (\_instance == null)              {                  \_instance = FindObjectOfType(typeof(T)) as T;                    if (\_instance == null)                  {                      Debug.LogError("There's no active " + typeof(T) + " in this scene");                  }              }                return \_instance;          }      }  } |

코드를 잘 보시면 <T>라고 되어 있는데, 나중에 사용할 때 T 대신에 클래스 이름을 넣어주면 됩니다.

사용하는 방법은 아래 스크립트 내용을 참고하시면 됩니다.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | using UnityEngine;  using System.Collections;    public class ManagerClass : Singleton<ManagerClass>  {    } |

쉽죠? MonoBehaviour 대신에 Singleton<ManagerClass>와 같이 < > 꺽쇠 괄호 안에 해당 클래스 이름을 넣어주면, 매니져 클래스 마다 프로퍼티를 적어줄 필요가 없습니다.

다른 스크립트에서 매니져 클래스에 접근하는 방법은 지난 번과 동일 합니다. ManagerClass.Instance. 방법으로 접근하면 됩니다.

내용 끝까지 읽어주셔서 감사합니다.

배너 클릭은 저에게 많은 힘이 됩니다.