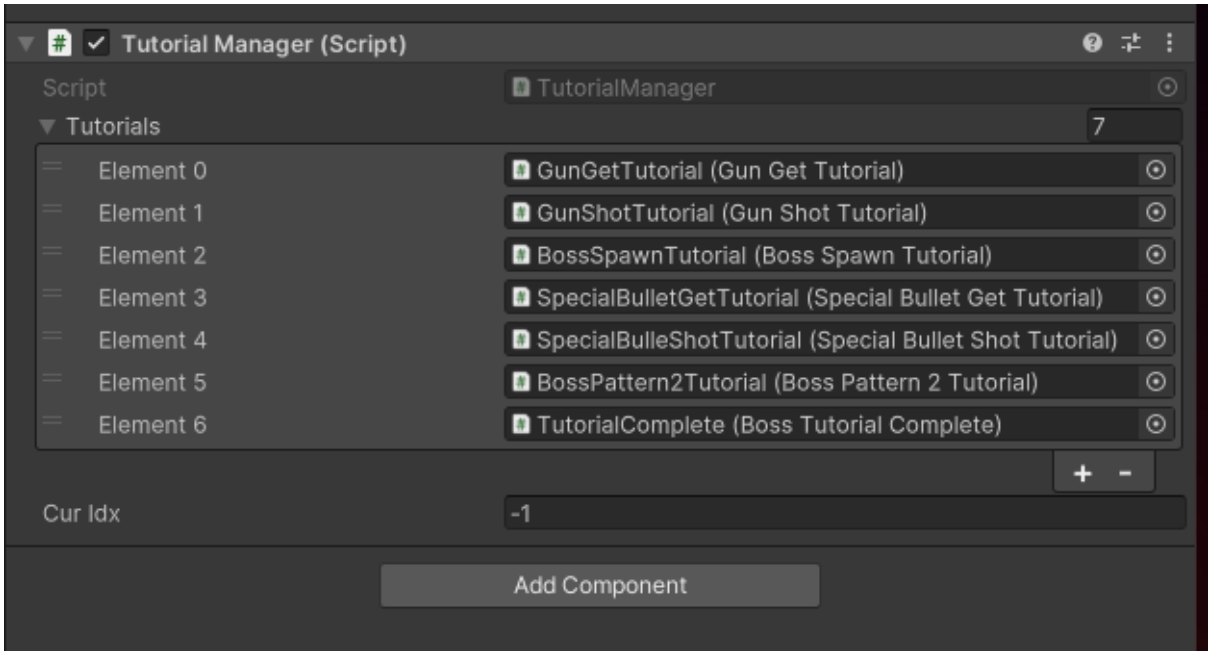
Tutorial System

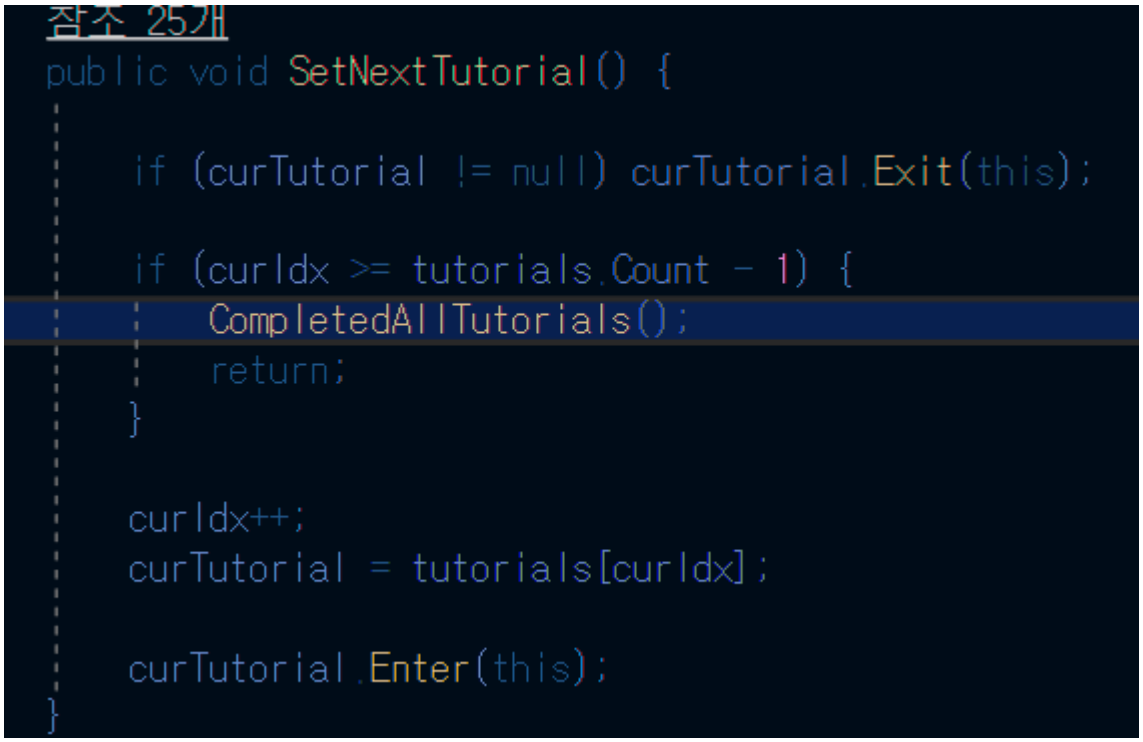
1. 설명

튜토리얼 시스템은 튜토리얼을 간단하게 구성하고 싶거나, 상황을 순차적으로 제어하고 싶을 때 사용하면 좋은 기능이다.

TutorialManager.cs

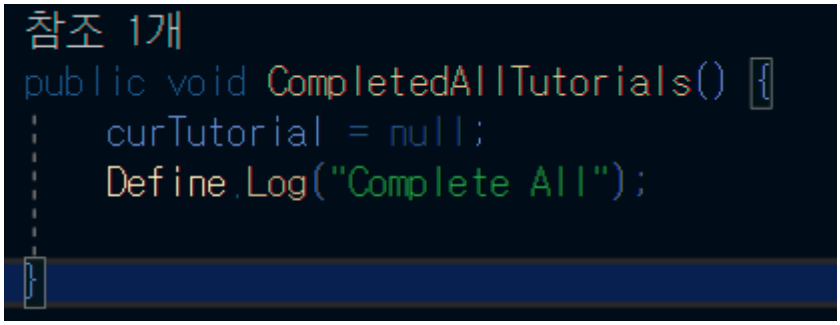


튜토리얼 상태를 순차적으로 관리해준다. 배열에 있는 순서대로 상황이 제어되며



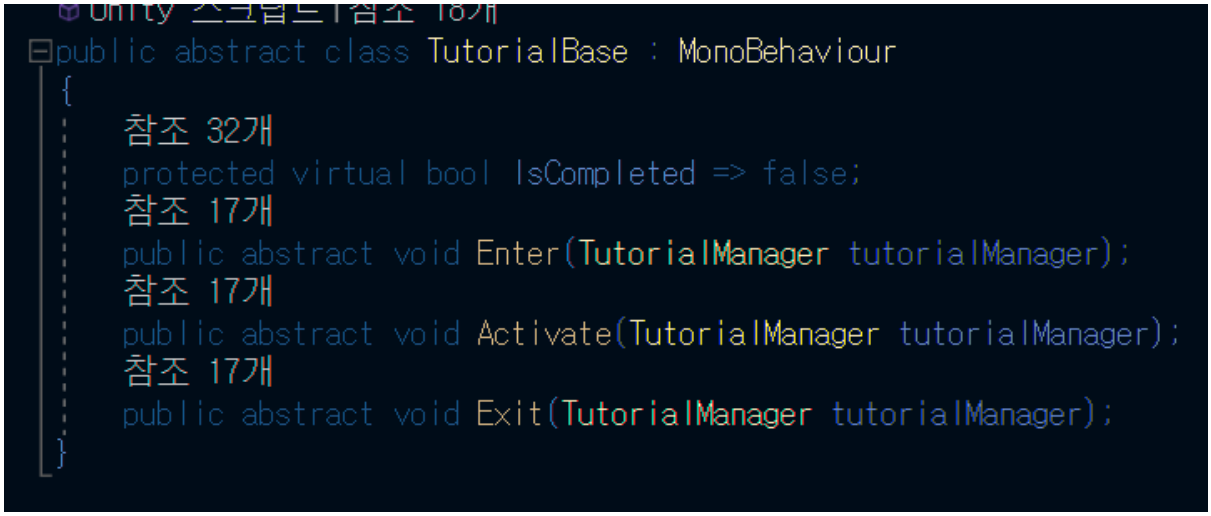
해당 메서드를 호출해 다음 상태로 넘어간다. 만약 처음 씬에 돌입하자마자 첫 튜토리얼을 진행하고 싶다면 바로 해당 메서드를 호출해주면 된다.

만약 남아있는 튜토리얼 객체가 없다면



해당 메서드가 호출되고 종료된다.

TutorialBase.cs



모든 튜토리얼 객체의 기본이 되는 추상 클래스다. 만들고 싶은 상황이 있다면 해당 클래스를 상속 받아서 구현해주면 된다.

IsCompleted

해당 튜토리얼이 종료되었는지 확인해주는 프로퍼티다.

public abstract void Enter(TutorialManager tutorialManager);

처음 해당 튜토리얼에 돌입할 때 호출되는 메서드. 이것에 초기화 작업을 한다.

public abstract void Activate(TutorialManager tutorialManager);

해당 튜토리얼이 진행되는 동안 TutorialManager의 Update에서 돌아가는 메서드. IsCompleted

가 true 가 되기를 기다리기도 하고, 현 상태에서 Update문에 들어갈 로직을 적어준다.

public abstract void Exit(TutorialManager tutorialManager);

해당 튜토리얼이 끝날 때 호출되는 메서드. 할당해둔 이벤트를 해제해 준다거나, null로 초기화해 주거나, 이벤트를 발생시키는 용도로 사용한다.

2. 사용법 (예시)

```
Unity 스크립트(자산 참조 1개) | 참조 0개
public class GunGetTutorial : TutorialBase {

    [Header("Tutorial Info")]
    [SerializeField] private DialogueGraph gunGetTutorialDialogue;
    [SerializeField] private Quest gunGetQuest;

    private Quest newQuest;

    참조 2개
    protected override bool IsCompleted => newQuest != null && newQuest.IsComplete;

    참조 2개
    public override void Activate(TutorialManager tutorialManager) {
        if (IsCompleted) {
            Define.Log("완료");
            tutorialManager.SetNextTutorial();
        }
    }

    참조 2개
    public override void Enter(TutorialManager tutorialManager) {
        Access.Player.StopPlayer();
        gunGetTutorialDialogue.BindEventAtEventNode("GunSpawnEvent", GunSpawn);
        Access.DialogueM.RegisterDialogue(gunGetTutorialDialogue);

    }

    참조 2개
    public override void Exit(TutorialManager tutorialManager) {
        gunGetTutorialDialogue.RemoveEventAtEventNode("GunSpawnEvent", GunSpawn);
    }

    참조 2개
    private void GunSpawn() {
        Access.BossStageM.GunSpawn();
        newQuest = Access.QuestM.Register(gunGetQuest);
        Access.Player.MovePlayer();
    }
}
```

위 스크립트는 총을 줍는 튜토리얼을 코드로 나타낸 것이다. 이렇게 원하는 스크립트를 만들어서, TutorialBase를 상속받아주고 추상 메서드와 가상 메서드를 구현해준다.

위 스크립트는 총을 줍는 퀘스트를 만들어서 해당 퀘스트가 클리어될 때 까지 대기해주는 역할을 해준다.

만약 튜토리얼이 클리어되었다면

tutorialManager.SetNextTutorial();

해당 메서드를 호출해줘, 다음 튜토리얼로 넘어간다.

또한 위 스크립트에서

Enter ->

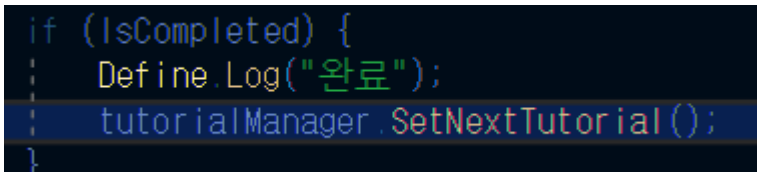
플레이어의 동작을 제한하고, 대화에 이벤트를 등록해준 뒤 대화를 등록한다.

해당 대화가 끝나면 GunSpawn 메서드가 호출되고 총이 스폰되고 플레이어는 다시 움직이게 되며 퀘스트가 등록된다.

해당 퀘스트가 완료되면

protected override bool IsCompleted => newQuest != null && newQuest.IsComplete;

이 부분이 true가 되어



여기로 넘어가게 된다.

이렇게 원하는 방식으로 상황을 제어해주면 된다.