

FadeUI Docs

23.08.06 손욱현

설명에 앞서

Action

- 함수를 변수처럼!
- 이벤트에 사용

Scriptable Object

- 공유되는 데이터 등을 파일로 저장

Action

변수는 함수의 호출에 따라 유동적으로 호출 가능

함수는 다른 함수 안에 직접 적어야 함

이 메서드가 실행되기
전 언제든지 변경 가능

5 20
 ↘
a = b + 10

```
public void Foo()  
{  
    Function();  
}
```

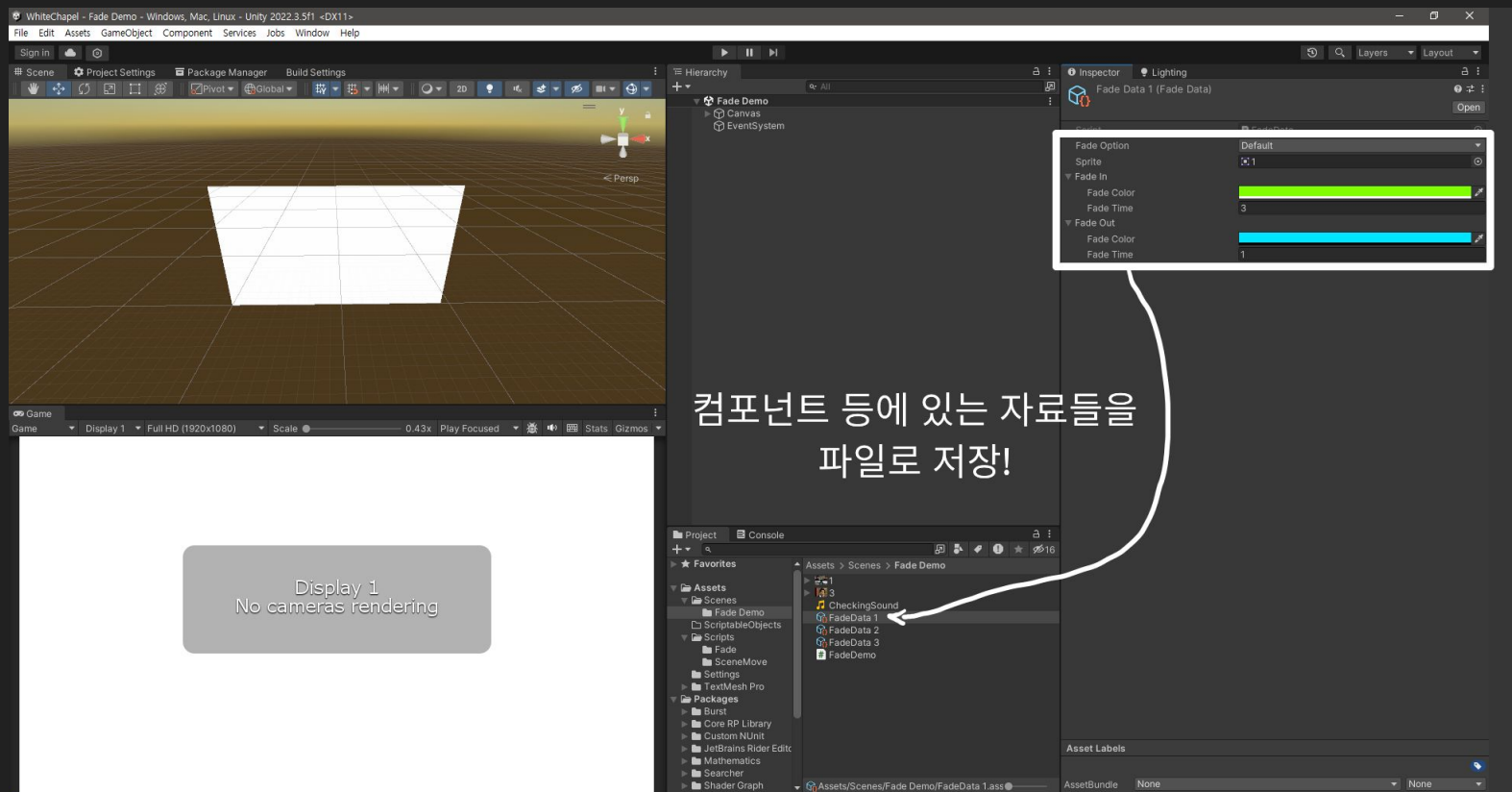
↖
변경할 수 없다

Action

함수도 변수처럼 사용!

```
Action action = new Action(Function1); // 함수 Function1() 저장  
action = new Action(Function2); // 함수 Function2() 저장  
action.Invoke(); // 저장된 함수 Function2() 실행
```

Scriptable Object

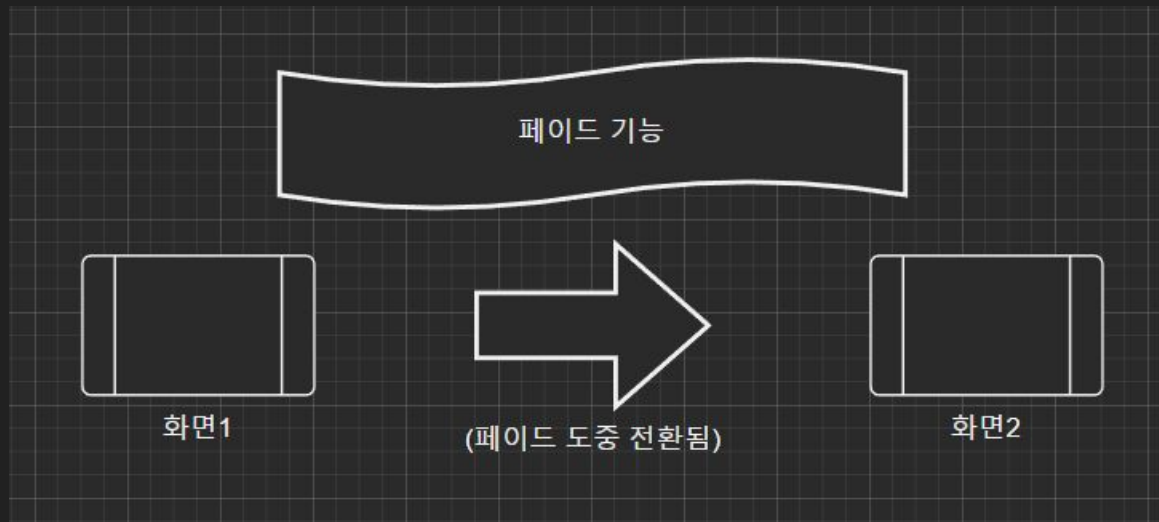


컴포넌트 등에 있는 자료들을
파일로 저장!

FadeUI의 목표

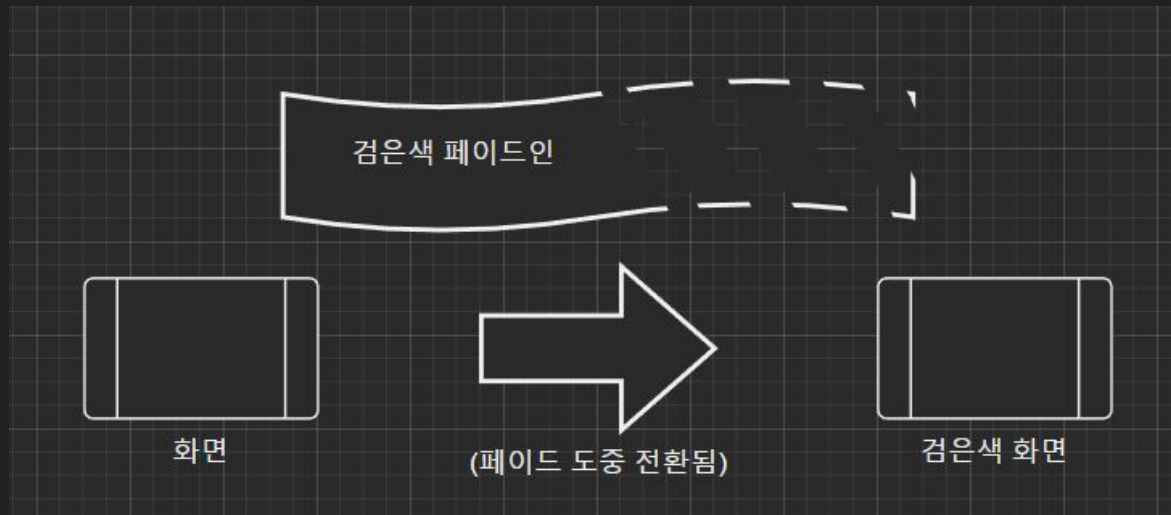
화면과 화면 사이의 전환

화면 1 > 페이드인 > 화면 전환 > 페이드아웃 > 화면2



FadeUI의 목표

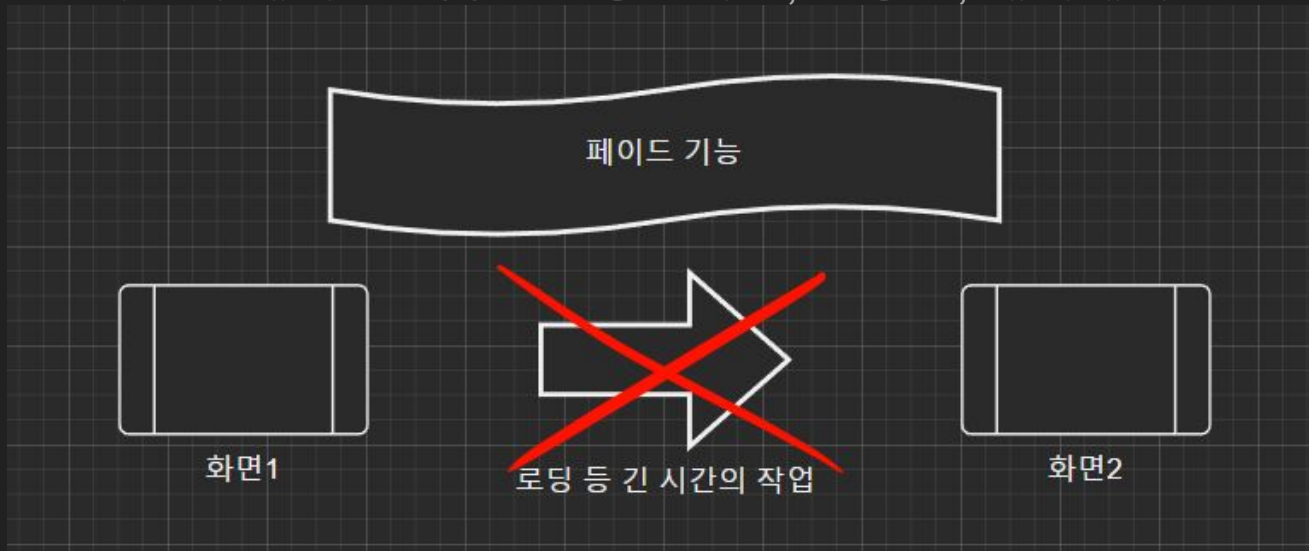
페이드 인과 결과화면이 비슷하다면, 페이드 아웃을 생략 가능
반대도 마찬가지



FadeUI의 목표

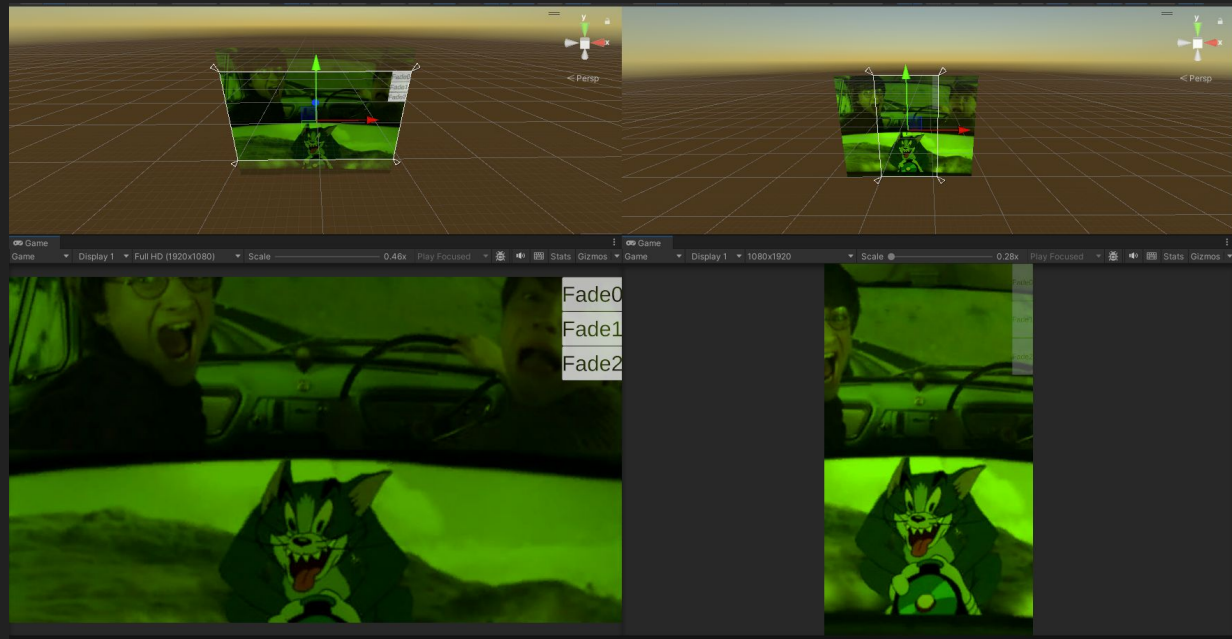
로딩과는 다르다 로딩과는!

긴 시간의 페이드아웃은 가능하지만, 로딩바, 애니메이션 같은 기능은 없음



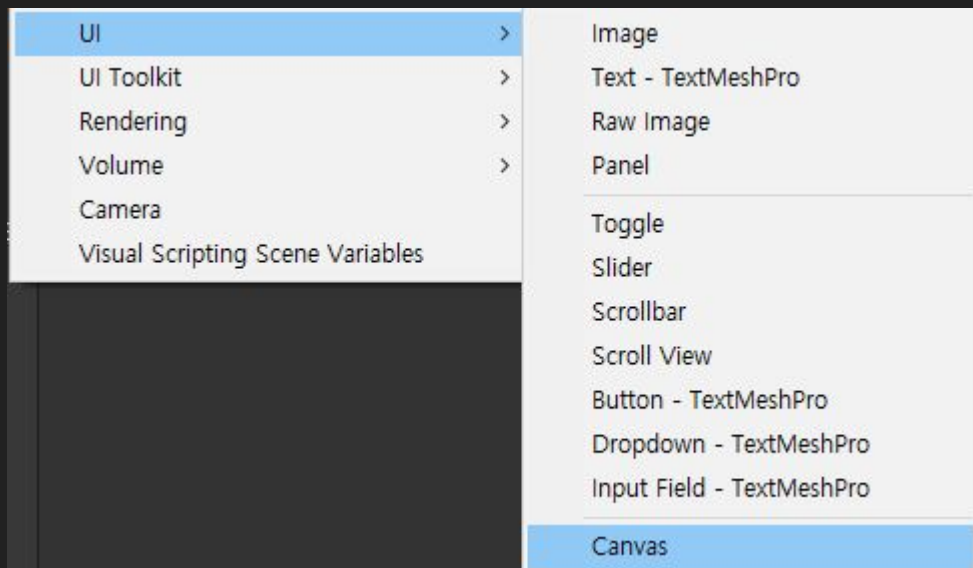
FadeUI의 목표

어떤 해상도에서든 화면 전체를 가리는것이 목표



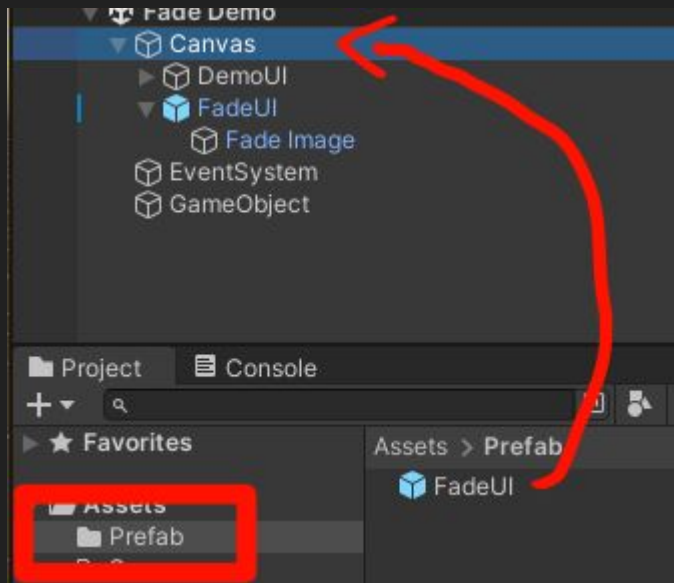
사용법 > 사전 세팅

하이어라키 우클릭 > UI > Canvas



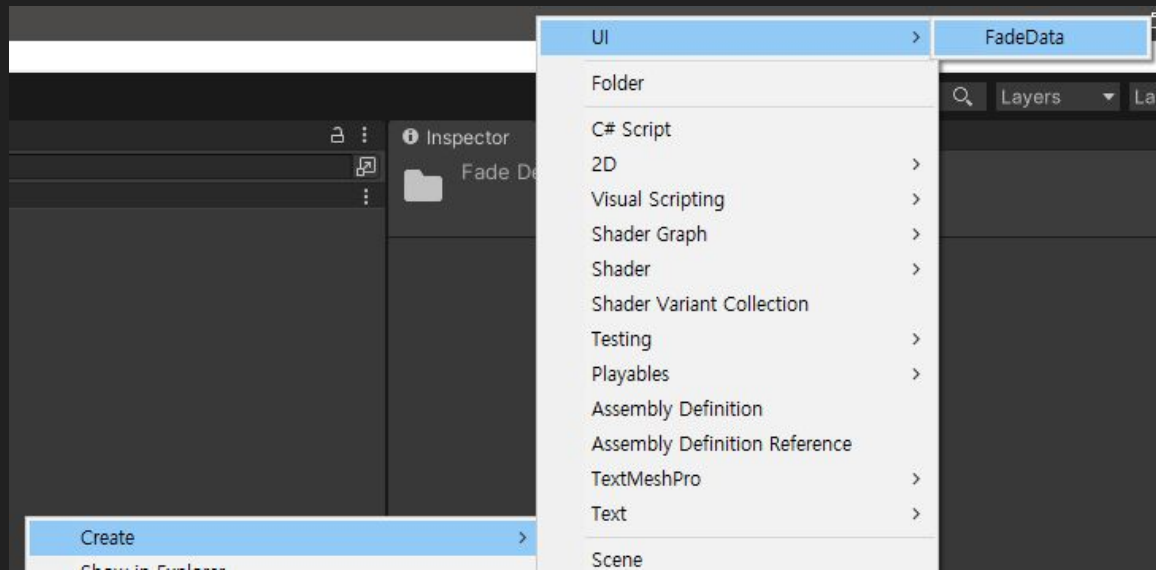
사용법 > 사전 세팅

사전에 준비된 Prefab을 Canvas의 자손으로 연결



사용법 > 디자이너의 경우

프로젝트 창에서 우클릭 > Create > UI > FadeData



사용법 > 디자이너의 경우

Fade Option 선택

- Default : 페이드 인, 페이드 아웃을 동시에 진행
- Clip Fade In : 페이드 인을 생략
- Clip Fade Out : 페이드 아웃을 생략



사용법 > 디자이너의 경우

프로젝트 창에서 우클릭 > Create > UI > FadeData

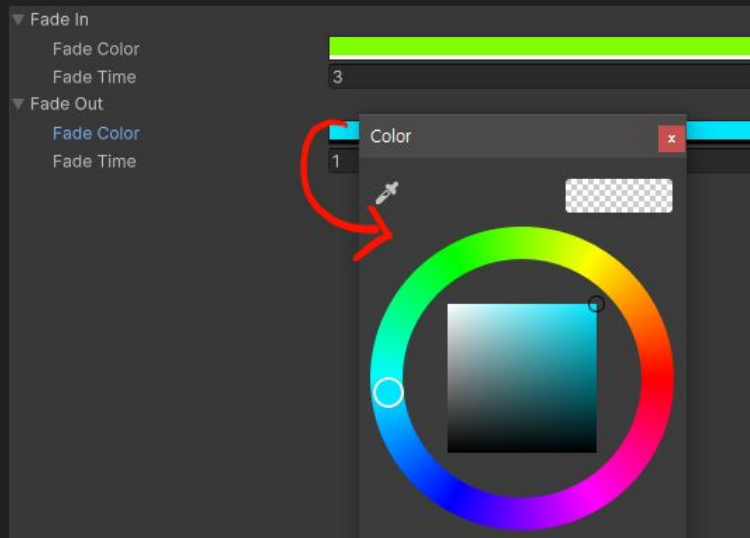
원하는 이미지를 스프라이트 형식으로 집어넣기



사용법 > 디자이너의 경우

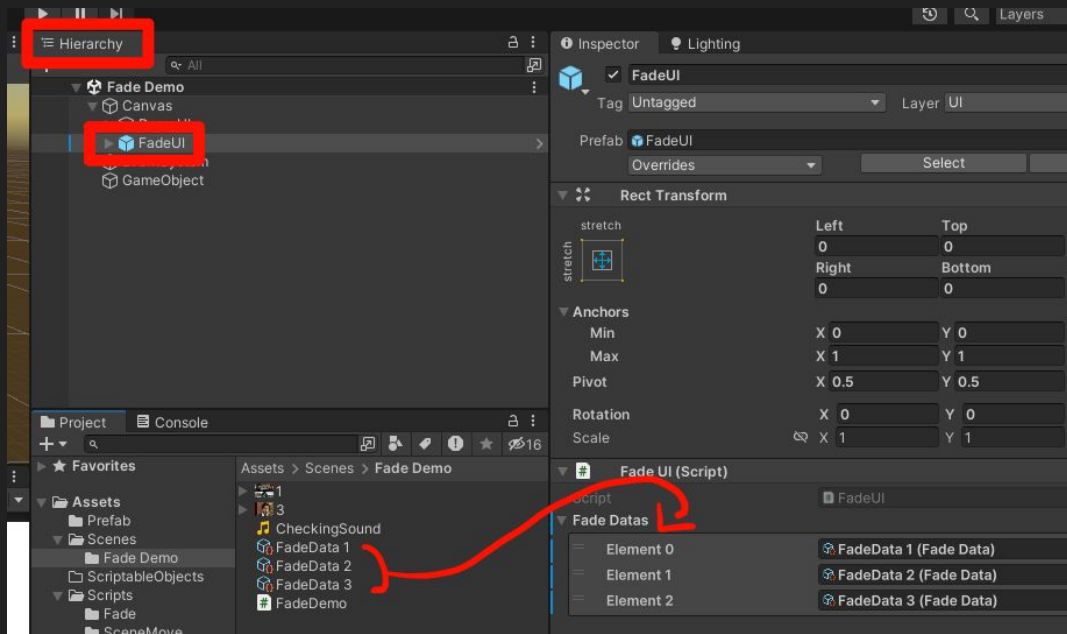
페이드 시간, 색을 결정

예시) 연두색으로 3초 페이드인, 하늘색으로 1초 페이드아웃



사용법 > 디자이너의 경우

준비된 스크립터블 오브젝트를 **하이어라키의** Fade Datas에 연결



사용법 > 프로그래머의 경우

FadeUI를 찾거나 연결 후 FadeStart 함수 호출

FadeMiddle과 같은 void 함수를 함수명만 작성하여 연결

어떤 페이드 인, 아웃 데이터를 사용할지 인스펙터에서 확인하고 작성



데모 씬

Assets > Scenes에서 데모 씬을 찾아 재생

