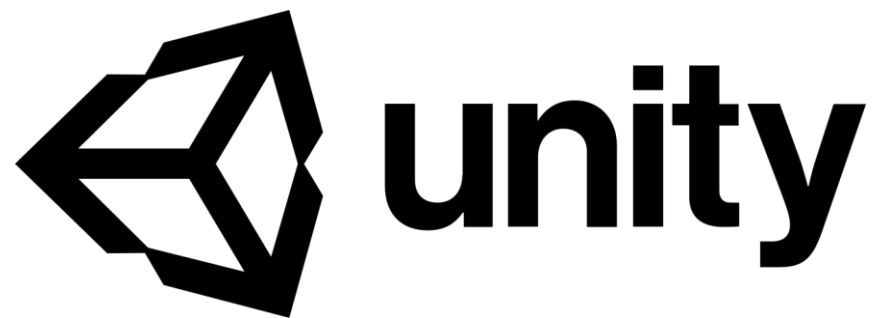


Asset 제작 : SBS게임 아카데미 선진송(SongTree)



AnimatorPro 1.0.0

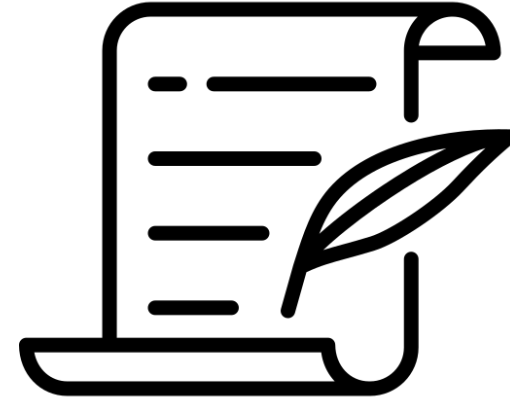
배포 : 김한용(NK Studio)

1.



개발된 이유

2.



사용 방법

개발된 이유

개발된 이유



AnimatorPro는 '선진송 개발자'님께서 개발을 해주셨습니다. 해당 Asset은 유니티의 경우 애니메이터 파라미터에 값을 전달 시, (SetFloat, SetInt, SetTrigger ...)와 같이 설정한 파라미터 자료형에 맞게 함수를 작성해야만 하는데요. 이 부분이 어찌보면 귀찮음이 발생하는 부분입니다.

그래서..!

해당 Asset에는 'SetParam'이라는 함수가 있으며 해당 함수를 사용으로 자료형의 구분없이 편하게 애니메이터의 파라미터에 값을 전달할 수 있습니다.


사용 방법

사용 방법



```
using UnityEngine.AnimatorPro;
```

1. 사용을 원하는 곳에서 해당 네임스페이스를 추가해줍니다.

```
private AnimatorPro AnimatorPro;  
public Animator anim;  GFX
```

2. AnimatorPro, Animator를 각각 전역변수로 선언해줍니다.

```
private void Awake()  
{  
    AnimatorPro = new AnimatorPro();  
  
    //사용할 애니메이션을 할당해주세요.  
    AnimatorPro.Init(anim);  
}
```

3. AnimatorPro의 객체를 생성해주고, Init에 사용할 Animator를 연결해줍니다.

사용 방법



4. `SetParam("Parameter Name/ID", (Int, Float, String) Value)`을 사용하여 애니메이터의 파라미터를 제어합니다.

(Value에 자료형을 넣으면, 자동으로 자료형의 타입을 판단하여 애니메이터의 파라미터에 값을 전달하게 됩니다.)

5. `SetTrigger("Parameter Name/ID")`만 예외적으로 `SetParm`을 사용하지 않습니다.

```
//이동 애니메이션 재생
var xx:float = Input.GetAxisRaw("Horizontal");
AnimatorPro.SetParam("Move", value: Mathf.Abs(xx));

//공격 애니메이션 재생
if (Input.GetKeyDown(KeyCode.Space))
    AnimatorPro.SetTrigger(paramName: "Attack");
```

사용 방법



파라미터는 Hash를 통한 ID값을 넣는 경우도 있습니다.

```
//해쉬를 사용하는 방법이 효율이 좋습니다.
```

```
private static readonly int ID_Move = Animator.StringToHash(name: "Move");
```

다음과 같이 파라미터를 전역 변수로 선언하고,

```
//이동 애니메이션 재생
```

```
var xx:float = Input.GetAxisRaw("Horizontal");
```

```
AnimatorPro.SetParam(ID_Move, value: Mathf.Abs(xx));
```

해당 ID 변수를 파라미터 네임부분에 넘겨주실 수 있습니다.