

게임의 몰입도를 높이는 카메라 제어 방법에 대한 연구

조성혜^o, 방정원^{*}

^o청강문화산업대학교 게임콘텐츠스쿨

e-mail: blackloup@naver.com^o, jwbang@ck.ac.kr^{*}

A study on the Camera Control Method to Increase the Immersion of Game

Seong-Hye Jo^o, Jung-Won Bang^{*}

^oSchool of Game, Chungkang College of Cultural Industries

● 요약 ●

수많은 사람들이 게임을 플레이할 때 마치 본인이 캐릭터가 된 듯한 이입을 하고 그를 통해 간접적으로 경험을 체험한다. 이때, 시각은 몰입하는데 빠질 수 없는 요소이며 게임에선 그 역할을 카메라가 대신하고 있다. 카메라를 통해 얻은 시각 정보로 현재 무슨 일이 일어나고 있는지, 지금 분위기가 어떤지를 판단하며 점점 게임에 몰입하게 되는 것이다. 캐릭터 간의 대화가 많은 추리 혹은 어드벤처 게임에서는 NPC와 대화하는 경우 실제로 대화하는 느낌이 들도록 카메라를 제어하는 것이 중요하다. 이 논문에서는 사람들이 쉽게 몰입할 수 있도록 카메라를 제어하는 방법을 연구하고, 엔진을 이용하여 추리 어드벤처 게임에 적용하였다.

키워드: 카메라 제어(Camera Control)

I. Introduction

시각은 사람이 상황을 판단할 때 가장 중요한 요소다. 게임에서 시각을 대신하는 카메라를 이용한다면 사람들이 게임에 쉽게 몰입할 수 있도록 도울 수 있을 것이다. 이를 위해 Cinemachine의 기능들과 Animation을 사용해 카메라의 움직임을 지정했으며 데이터의 값을 통해 작동할 수 있도록 구현했다.

II. Related works

1. Cinemachine

Cinemachine Brain 클래스는 Cinemachine Virtual Camera 와 Unity Camera 사이를 연결해 주는 클래스로, Cinemachine Brain 컴포넌트를 카메라에 추가해 Cinemachine Virtual Camera를 통해 이동할 카메라가 이 카메라임을 지정할 수 있다. 또한 Default Blend 요소를 통해 카메라 이동 시 어떤 방식으로 이동할지 지정할 수 있다. [1]

Cinemachine Virtual Camera 클래스는 빈 오브젝트에 추가해 메인 카메라가 이동할 위치와 각도를 정할 수 있으며 Priority 요소를 통해 이동의 우선순위를 정할 수 있다. Look At 요소를 통해서 카메라가 무엇을 중심으로 돌지를 정할 수 있으며 Follow를 통해 추가한 오브젝트가 움직일 때 카메라가 따라가는 효과를 낼 수 있다.[2]

2. Animation

Animation 컴포넌트는 유니티 애니메이션 시스템을 이용해 해당 오브젝트가 움직일 수 있게 지원한다. Animation 항목에 넣은 애니메이션을 사용해 해당 오브젝트가 움직일 기본 애니메이션을 지정할 수 있고 Animations에 몇 개의 애니메이션을 넣는지에 따라 Script로 연결할 수 있는 애니메이션의 개수가 늘어난다. 또한 Play Automatically를 체크해 오브젝트 활성화와 동시에 애니메이션을 작동시킬 수 있다.[3]

III. Implementation

Cinemachine Virtual Camera 기능을 사용해 카메라가 이동할 위치와 각도를 미리 지정해두고 animation 기능을 사용해 오브젝트가 활성화 될 때 동시에 애니메이션이 작동할 수 있도록 지정해 줄 수 있다. 하지만 하나의 씬 내에서 사용되는 카메라 제어는 한 가지가 아니며, 원하는 타이밍에 원하는 방식의 카메라 제어가 작동해야 하므로 추가적인 처리가 필요하다.

첫째, 해당되는 모든 카메라 제어 오브젝트에 Action 태그를 할당하여 씬이 변경될 경우 해당 태그의 오브젝트들만 제어할 수 있도록 한다.

```

actionCamera = GameObject.FindGameObjectsWithTag("Action");
for(int i=0; i<actionCamera.Length; i++)
{
    actionCamera[i].SetActive(false);
}
    
```

미리 GameObject 배열을 만들어 씬을 넘어갈 경우 위와 같은 메소드를 이용해 Action 태그의 오브젝트들을 찾아 저장해 비활성화시킨다.

둘째, 데이터의 CamCtrl 항목의 내용을 받아와 저장해둔 오브젝트들의 이름과 비교해 같은 이름의 오브젝트를 활성화시킨다.

```

public void StartActionCam(string cameraName)
{
    if (_action != null)
    {
        _action.SetActive(false);
        _action = null;
    }
    for (int i = 0; i < actionCamera.Length; i++)
    {
        if (actionCamera[i].name == cameraName)
        {
            _action = actionCamera[i];
            _action.SetActive(true);
            break;
        }
    }
}
    
```

이런 식으로 해당 오브젝트를 활성화시키면 Cinemachine Virtual Camera 컴포넌트로 인해 메인 카메라가 해당 오브젝트의 위치와 각도에 따라 이동하고 Animation 컴포넌트에 의해 애니메이션이 작동되면서 카메라를 원하는 대로 제어할 수 있다.

IV. Conclusions

Cinemachine Brain/ Cinemachine Virtual Camera 와 Animation 기능을 사용해 플레이어의 몰입을 높일 수 있게 도와주는 카메라 제어를 구현해보았다.

게임을 하면서 얼마나 그 속에 몰입할 수 있는가는 중요한 부분이다. 몰입도에 따라 게임을 통해 무엇을 느끼고 얼마만큼 느끼는지가 크게 달라지기 때문이다. 세상의 모든 일을 체험해 볼 수 없기에 게임을 통한 간접 체험은 중요해질 것이고, 그런 만큼 플레이어의 몰입을 높일 수 있는 방법에 대한 연구가 계속되어야 한다.

REFERENCES

- [1] <https://docs.unity3d.com/Packages/com.unity.cinemachine@2.1/api/Cinemachine.CinemachineBrain.html>
- [2] <https://docs.unity3d.com/Packages/com.unity.cinemachine@2.1/api/Cinemachine.CinemachineVirtualCamera.html>
- [3] <http://docs.unity3d.com/Manual/class-Animation.html>