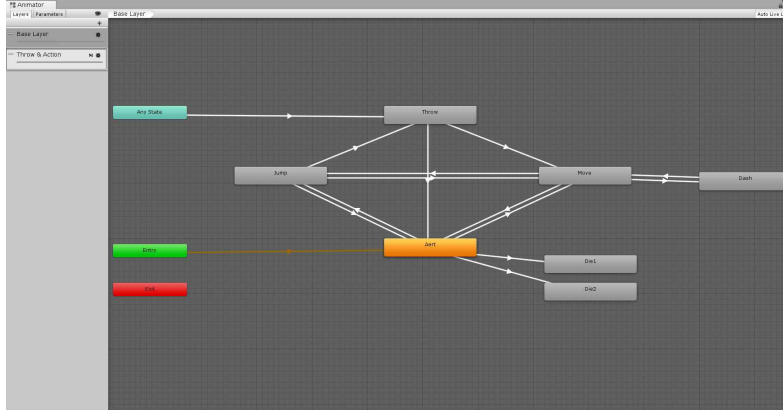


2018-2학기 세종창의학기제 주간학습보고서

이름	오민규	학과(전공)	컴퓨터공학과		
학번	128206	학년	4		
연락처	010-8843-4757	e-Mail	hotalsrb12@naver.com		
과목명	자기주도 창의전공 I, II, III, IV	신청학점	12	분반	3
학습기간	2018. 09. 17 ~ 2018. 09. 23	학습주차	3	학습시간	62
창의과제	네트워크를 통해 상대방과의 실감나는 대전형 슈팅 3D 모바일게임 개발				
금주 학습목표	<ul style="list-style-type: none"> - 캐릭터 애니메이션 매니저 구현 - 캐릭터 애니메이션시간 컨트롤러 구현 및 싱크 - 유니티 네트워크(UNET) 학습 				
학습내용	<p>1. 캐릭터 애니메이션 매니저 구현</p> <p>2주차에서 설계했던 캐릭터 애니메이션 매니저를 토대로 애니메이터를 구현하였다.</p> <p>캐릭터가 움직이며 점프했을 경우, 움직이다가 회피를 했을 경우 등등 많은 경우의 수가 존재하기 때문에 그런 설계도를 바탕으로 그래프를 만들었고 각 애니메이션 마다 트리거 변수를 생성하여 해당 트리거가 발생하였을 때 실행되도록 제작하였다. 움직이며 공격을 할 경우 캐릭터의 상체와 하체를 분리시켜야하므로 애니메이터에서 다른 Layer를 생성하여 애니메이션을 별도로 제작하였다. 이동하면서 공격할 경우, 하체에는 이동 애니메이션을 적용하였고 상체에는 공격 애니메이션을 적용하여 원하던 애니메이션을 구현하여 적용시킬 수 있었다.</p>  <p>애니메이터 제작을 완료한 후, 애니메이션 매니저 클래스를 생성하여 하나의 클래스로 모든 플레이어들과 적 캐릭터들이 공통으로 사용할 수 있게 제작하여 재사용성을 증가시켰다. 그리고 캐릭터의 기능과 애니메이션을 연결시켜 게임플레이를 하였을 때 두 구성요소들이 같이 적용되는 것을 확인할 수 있었다. 하지만 공격키를 두 번 연속으로 눌렀을 때, 첫 번째 애니메이션이 끝까지 실행되기 전에 애니메이션이 다시 실행되는 문제가 발생하였다. 이는 기능과 싱크를 맞추지 않으면 해결할 수 없다는 것을 알게 되어 기능과 애니메이션 싱크를 맞추는 작업을 해야 한다.</p>				

2. 캐릭터 애니메이션 시간 컨트롤러 구현

특정 기능이 한번 발생하는 동안 애니메이션이 한번 실행되어야한다. 하지만 매니저를 구현하고 적용시켰을 시 싱크 문제가 발생하여 이를 해결하기 위해 애니메이션 타임 컨트롤러 클래스를 생성하여 각 행동의 한번 실행되는 애니메이션 시간을 변수에 저장시켰다. 저장된 시간을 가지고 기능과 애니메이션이 한번 발생하는 로직을 구현하였으며 게임이 시작되었을 때 싱글톤 패턴을 사용하여 하나의 객체를 생성하고 사용함으로써 메모리에 한번 적재시켜 사용하도록 구현하여 재활용성과 불필요한 생성을 막아 효율적으로 프로그램이 실행되도록 제작하였다.

```
public class AnimationTimeController : MonoBehaviour<AnimationTimeController>
{
    // 애니메이션 타입(Enum 변수)
    public enum AnimeType { ALERT = 0, MOVE, JUMP, DASH, THROW, DIE, DIE2 }

    // 애니메이터 변수
    public Animator anim;

    // 애니메이터 컨트롤러
    private RuntimeAnimatorController animCtrl;

    // 애니메이션 시간
    private float[] animTimes;
    public float this[int index]
    {
        get { return animTimes[index]; }
    }

    // 애니메이션 중복여부
    private bool[] isAnimOverlaps;

    // 애니메이션 이름
    private readonly string[] animNames = { "Alert", "Move", "Jump", "Dash", "ATK1", "Die1", "Die2" };

    private void Awake()
    {
        // 모노싱글톤 애니메이션 컨트롤러 할당
        Instance = this;

        // 애니메이션 컴포넌트 할당
        anim = GetComponent<Animator>();
    }
}
```

아직 모바일 기기에서 테스트를 해보지 못하였지만 기기 성능에 따라서 시간으로 제어하는 경우 문제가 발생 할 수 있다고 생각한다. 이 문제가 발생하였을 시 애니메이션 프레임으로 제어하는 방법을 고안해보거나 교수님께 도움을 요청하여 해결해야겠다.

3. 유니티 네트워크(UNET) 학습

유니티 네트워크 API(UNET)을 이용하여 네트워크를 구현하고자한다. 이전에 윈도우 소켓을 이용하여 게임을 개발한 경험이 있지만 유니티에서 네트워크를 구현하기 위해 API를 학습하고 적용시켜야한다. API를 사용하여 몇 차례 게임을 개발한 경험 덕분에 학습하는데 많은 시간을 사용하지 않고 적용할 수 있을 것이라고 생각된다. UNET이 작동되는 로직에 관하여 공부하였고 네트워크를 적용시키기 위해서 프로젝트 세팅을 마무리하였다.

4. 느낀점

지난주 캐릭터가 움직이면서 카메라가 떨리는 현상이 발생한 부분을 해결하였다. 캐릭터가 매 프레임을 FixedUpdate(물리)에서 처리를 하며 움직이는데 카메라는 LastUpdate에서 수정을 하였던 것이 문제였다. 카메라도 마찬가지로 FixedUpdate에서 처리를 해주었더니 그 문제점이 해결이 되었다. 하지만 FixedUpdate에서 이렇게 많이 처리를 하는 것은 좋지 않은 방법으로 알고 있다. 1프레임에 Update함수나 LastUpdate함수는 1번 돌아가지만 지금 처리하고 있는 FixedUpdate는 고정된 프레임율의 프레임에 따라 호출이 되기 때문에 Update 함수보다 더 많이 호출되는 경우가 생기가 된다. 그 만큼 기기에서 처리해야하는 양이 몇 배 더 많아지기 때문에 게임 성능의 저하를 나타낼 수 있다. 개발을 하며 성능 테스트를 하는 과정에서 문제가 발생한다면 다른 방법으로 처리 하는 방법을 교수님께 자문을 통하여 얻어야 한다고 생각된다. 이전에 게임을 제작할 때 매니저를 사용하지 않고 객체에 스크립트를

	<p>넣어서 Prefab화 시키고 사용하여 재활용성을 무시한 채 사용하였지만 이번에 매니저를 사용함으로써 하나의 스크립트로 여러 가지 객체들을 제어할 수 있다는 점을 배우고 적용시킴으로써 배움을 얻고 개발실력 또한 향상되었다고 생각한다. 캐릭터 싱크를 맞추는 점에서 고민을 많이 하였다. 캐릭터 애니메이션이 한번 실행되는 예상 시간을 잡아 사용할지, 애니메이션이 발생하는 트리거 또는 bool 값에 의해 제어할지 고민을 많이 하였다. 애니메이션이 재생되는 시간을 유니티에서 제공해주기 때문에 “이 방법을 가지고와서 애니메이션과 기능을 제어하면 되겠다”라고 생각하였다. 이게 옳은 방법인지는 잘 모르겠다. 구글링을 통하여 검색을 하더라도 이 부분에 대해서는 마땅히 정답이라는 방법이 없어서 생각한대로 구현한 것이다. 이 방법을 사용하여 개발하였을 때 코드의 길이는 많아 졌고 게임 플레이 중에 한번만 메모리에 올려 사용한다는 점에서 사용하였다. 이번 주 학습을 하며 많은 것을 배운 것 같다. 매니저, 시간 컨트롤러를 싱글톤을 사용하여 메모리에 한번 적재 시키는 점과 UNET이라는 API를 공부하고 프로젝트 세팅을 하며 네트워크를 적용해보았다는 점에서 내가 기존에 알던 지식으로 개발을 하는 것이 아닌 새로운 것을 배우고 구현하며 적용시키는 과정이 뜻 깊은 한 주였다.</p>
참고자료 및 문헌	<ul style="list-style-type: none"> - 절대강좌! 유니티 2018 - https://docs.unity3d.com/kr/2018.1/Manual/UnityManual.html (유니티 공식 메뉴얼)
학습방법	<p>책과 유니티 공식 홈페이지의 매뉴얼을 참고하며 UNET에 대해 학습하였고 프로젝트에 네트워크 세팅을 완료하였다. 애니메이터를 제작할 때 필요한 요소들을 생성하는 방법과 Layer를 분리하는 방법을 문헌에서 도움을 얻었다.</p>
학습성과 및 목표달성도	<ul style="list-style-type: none"> - 캐릭터 애니메이션 매니저 구현 완료(100%) - 캐릭터 기능과 애니메이션 싱크(80%) - 프로젝트 네트워크 세팅(100%)
내주 계획	<p>UNET에 대해 더 많이 학습하고 학습한 내용을 프로젝트에 적용시킬 것이다. 프로그램을 빌드한 후 두 개의 게임을 실행하였을 때 내 캐릭터와 상대방의 캐릭터가 독립적으로 움직이며 이전까지 구현한 기능들이 네트워크를 통하여 송·수신이 제대로 작동하도록 구현해야 한다.</p>

2018. 09. 21 .

지도교수

(인)