

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. 11. 26 ~ 2018. 12. 02	학습주차	12	학습시간	65
창의과제	네트워크를 통해 상대방과의 실감나는 대전형 슈팅 3D 모바일게임 개발				
금주 학습목표	<ul style="list-style-type: none"> - FSM(Finite State Machine) 설계 - FSM(Finite State Machine) 구현 및 유니티 NavMesh와 연동 - 초보 난이도의 AI가 적용된 싱글플레이 테스트 - 세종 ICT 창업 아이디어 경진대회 참가 				
학습내용	<p>1. FSM(Finite State Machine) 설계</p> <p>처음에 게임을 기획하였을 때 싱글플레이에서 초보 / 중수 / 고수의 AI 난이도를 만들고자 하였다. 초보와 중수 난이도의 AI는 FSM을 이용하여 개발하고 고수는 유니티 머신러닝 에이전트를 활용하여 개발할 예정이다. FSM은 AI가 어떤 상태를 가지고 그 상태에 대한 행동을 실행하고 있을 때 어떤 이벤트가 발생한다면 이벤트에 맞는 상태로 전이시키는 알고리즘이다. 이전에 게임 개발을 하며 이 알고리즘을 직접 만들었던 적이 있었는데 그때는 RPG게임을 개발하여 여러 개의 상태와 이벤트들을 정의하고 연결지어 사용하였다. 하지만 지금 개발하는 게임은 대전게임이기 때문에 AI가 상대팀을 발견하고 쫓아가며 공격을 한다면 단순하고 게임의 예측이 가능 할 것이다. 대전 게임에서는 상태가 많이 있다면 오히려 사용자들이 쉽게 게임을 할 수 있을 것이라고 생각하여 최대한 적은 상태와 적은 이벤트를 가진 FSM을 제작하기로 하였다. 처음엔 다섯 가지 상태를 가진 AI를 설계하였다. 유희, 공격, 추적, 수색, 사망 상태로 어떤 상태에서 상대팀 캐릭터들이 모두 사망하는 이벤트가 발생한다면 유희 상태로 전이되고 상대팀 캐릭터들이 리스폰 된다면 수색을 시작한다. 수색 중 AI의 영역에 상대팀이 있다면 추적을 하고 추적 중 공격 범위 내에 상대팀이 있다면 공격을 한다. HP가 0 이 된다면 사망 상태로 변경하게 된다. 이렇게 설계를 하였지만 만약 어떤 물체가 그 사이를 가로막고 있고 공격범위 내에 있다면 AI는 방향이나 건물을 뚫으려고 할 것이다. 이런 문제 때문에 추적은 좋지 않은 상태이므로 제거하였고 유희상태의 경우 AI 캐릭터들이 멈추어 버리게 된다면 플레이어는 상대 AI의 위치와 다음 이동하는 방향을 알 것이다. 따라서 유희 상태와 추적을 제거하였다. 따라서 최종 설계는 공격, 수색, 사망 상태 3개의 상태로 이루어져 있으며 게임 예측을 피하기 위해 AI가 사망했다면 Way Point를 새롭게 설정하여 AI가 항상 다르게 움직이게 하였다. 공격과 수색 상태는 이전에 설계했던 것을 그대로 적용하였다.</p>				

2. FSM(Finite State Machine) 구현 및 유니티 NavMesh와 연동

설계한 FSM을 가지고 게임에 적용 시켜야한다. Switch-Case문을 사용하게 된다면 나중에 중급 난이도의 AI를 개발할 때 문제가 될 것이라고 생각하여 초기에 디자인 패턴을 공부하였을 때 학습하였던 State Design 패턴을 적용하여 새로운 상태가 추가될 때마다 스위치-케이스 문에 넣는다면 계속 코드를 작성하여 추가하여야 하므로 적용하였다. 현재 상태와 현재 이벤트를 관리하는 Finite State 클래스(FSM 기능)와 State 디자인 패턴을 적용하여 STL인 Dictionary를 사용하여 상태와 이벤트를 저장하는 변수와 상태가 추가되거나 현재 상태에서 어떤 이벤트가 들어왔을 때 전이되는 상태를 출력하는 Finite State Machine 클래스를 적용하였고, 수색, 공격, 사망 상태 클래스를 정의하여 각 상태에 따라 기능을 구현하였다. 이전에 명령패턴에 적용한 Actor를 기반으로 실질적인 기능을 다시 구현하지 않아도 되었으며 세부적인 내용들만 추가하여 코드를 작성하였다. FSM을 다 구현하고 AI가 갈 수 없는 지역과 Way point를 제작하기 위해서 NavMesh 기능을 적용시켜 Way point를 맵에 적절한 위치에 찍어서 사용하였다.

```
void AddStateTransition(DWORD state, DWORD iEvent, DWORD outState)
{
    std::map<DWORD, FiniteState*>::iterator iter;
    FiniteState *pState = NULL;
    iter = m_mapState.begin();
    while (iter != m_mapState.end())
    {
        pState = iter->second;
        if (pState->m_dwState == state)
            break;
        iter++;
    }
    // 만일 동일한 State가 존재하지 않는다면 새로 생성한다.
    if (iter == m_mapState.end())
    {
        pState = new FiniteState(state, iEvent);
        m_mapState[state] = pState;
    }

    // 상태 전이 정보를 추가한다.
    if (pState != NULL)
    {
        pState->AddTransition(iEvent, outState);
    }
}

DWORD GetOutputState(DWORD iState, DWORD iEvent)
{
    return m_mapState[iState]->OutputState(iEvent);
}
```

3. 초보 난이도의 AI가 적용된 싱글플레이 테스트

FSM 구현을 완료하여 유니티 NavMesh와 연동을 하고 게임이 잘 진행되는지 AI의 상태가 잘 바뀌고 그 기능을 제대로 하는지 확인하는 테스트 과정을 거치며 발생하는 문제를 해결해야한다. 처음에 테스트를 진행하였을 때 AI의 수색은 정상적으로 작동을 하였지만 공격 범위에 상대팀의 캐릭터가 2명이상 존재하였을 때 Rotation 부분에서 문제가 발생하였다. 현재 사용하는 알고리즘이 경계 안에 존재한다면 리스트에 추가를 하고 처리를 하는데 리스트에 존재하는 첫 번째 대상을 바라보고 공격을 수행하게 된다. 하지만 1 프레임에 1번 씩 함수를 호출하기 때문에 2명이 경계 안에 존재하게 된다면 AI는 2명을 번갈아가며 쳐다보기 때문에 계속해서 회전을 수행한다. 따라서 이 문제를 해결하기 위해서 공격 기능을 구현하는데 사용하였던 Throw delay time을 적용 시켜서 공격이 가능할 때만 해당 캐릭터를 쳐다보고 공격을 하도록 수정하였다. 그리고 캐릭터가 사망상태라면 Way point를 재 설정해야하는데 너무

	<p>단순하게 point가 변경되어 이동 패턴의 큰 변화가 존재하지 않았다. 현재 Way point는 그래프 형태로 구축되어 있다. 이 문제를 해결하기 위해서는 이전 수행한 그래프의 이동 정보를 가지고 있어야 한다고 판단되어 항상 스폰 될 때 그래프 정보를 저장시키고 사망하였을 때 이전 그래프의 이동 순서를 가지고 겹치지 않게 Random으로 다시 이동 패턴을 생성하여 이 문제를 해결하였다.</p> <p>4. 세종 ICT 창업 아이디어 경진대회 참가</p> <p>지난주에 사업계획서와 발표 자료 작성하였다. 금요일에 경진대회가 진행되므로 틈틈이 시간이 날 때 발표 연습을 진행하였다. 17팀이 본선에 진출해있었고 발표순서가 12시로 잡혀있었다. 발표하기 좋은 시간이었다. 하지만 발표에 스킬이 부족하고 발표가 가장 취약했던 부분 중 한 부분이었기 때문에 걱정이 많이 되었다. 그래서 항상 쉬는 시간이 있을 때마다 계속해서 연습을 진행하였다. 금요일이 되고 발표를 시작하였다. 첫 번째 팀부터 좋은 아이디어가 나오기 시작하였다. 계속해서 발표를 진행하는데 나는 조금 위축이 되어있었다. 다른 팀들은 대부분 실생활에서 필요한 아이디어를 출시하였다. 내가 낸 아이디어도 충분히 좋다고 생각하였지만 아무래도 오락이라는 요소가 들어가 있기 때문에 소극적이 된 것 같다. 발표가 계속 진행이 되고 내 차례가 되었다. 발표 연습을 그래도 틈틈이 진행했기 때문에 평소 다른 발표를 할 때보다 자연스럽고 뜻하는 바를 제대로 전하였다. 발표를 잘 마무리하고 질의응답 시간에서 오히려 좋은 평가를 받게 되었다. 기존 게임들의 문제점들을 잘 파악한 덕분이었다. 쉬는 시간을 가지고 마지막 4팀의 발표를 진행을 하였고 마무리가 되었다. 이제 시상식이 진행될 차례가 왔다. 좋은 평가를 받았기 때문에 수상에 기대를 하였다. 시상식이 진행되자마자 바로 팀 이름이 호명되어 우수상을 받는 좋은 결과를 얻게 되었다. 수상을 하고 내가 왜 수상을 하였는지 생각해보았다. 이유는 간단하였다. 다른 팀들의 좋은 아이디어들이 많이 쏟아져 나왔지만 이 대회는 창업 아이디어 경진대회이다. 우리 팀의 아이디어도 좋은 아이디어이지만 이전에 정부 사업을 진행하며 준비했던 덕분에 창업을 하였을 때 사업에 대한 방향성과 전략 등이 다른 팀에 비해 잘 정리되어 있었기 때문에 수상을 하였던 것이라고 생각했다. 한 가지 더 느낀점은 자신감을 더 키워야한다고 생각했다. 내가 생각한 아이템이 어떤 것보다 부족하다고 생각할 필요가 없기 때문이다. 시상식을 마치고 사진촬영을 하고 경진대회를 마무리 하였다.</p>
참고자료 및 문헌	<ul style="list-style-type: none"> - 절대강좌! 유니티 - 구글링 - https://docs.unity3d.com/kr/2018.1/Manual/UnityManual.html (유니티 공식 메뉴얼)
학습방법	구글링을 통하여 FSM(Finite State Machine)을 효율적으로 설계하는 방법을 찾아보고 설계

	를 한다. 설계를 바탕으로 스크립트를 생성하여 구현을 한다. 절대강좌! 유니티 문헌과 유니티 공식 메뉴얼을 통해서 NavMesh 사용법과 변수, 기능을 학습하고 구현한 FSM(Finite State Machine)과 연동을 시작한다. 경진대회를 위한 발표 자료를 가지고 발표 연습을 진행한다.
학습성과 및 목표달성도	<ul style="list-style-type: none"> - FSM(Finite State Machine) 설계(100%) - FSM(Finite State Machine) 구현 및 유니티 NavMesh와 연동(100%) - 싱글플레이 테스트(90%) - 세종 ICT 창업 아이디어 경진대회 우수상 수상
내주 계획	플레이어가 공격할 때 마다 아이스크림 볼 객체를 생성하였는데 이는 많은 메모리를 사용하여 가비지 콜렉터가 하는 일이 많아져서 게임 성능의 문제를 발생한다. 따라서 오브젝트 풀을 구현하고 이를 관리하는 매니저를 구현한다. 그리고 프로토타입 맵을 제작하여 프로토타입 게임이 나올 수 있게 생성한다.

2018. 12. 03 .

지도교수

(인)