|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **주차** | 14주차 | **기간** | 2023.10.03~ 2023.10.09 | **지도교수** | (서명) |
| 이번주 한 일 요약 | 별도의 에어 컴벳 장르 게임을 제작해보면서 언리얼 학습  나이아가라 시스템 학습  인프런 게임서버 멀티쓰레드 강의 학습 시작 | | | | |

<상세 수행내용>

# 직접 게임 하나를 개발하는 것이 언리얼 엔진을 학습하는 데에 도움이 되고, 나아가 긍정적인 자신감을 얻는 데에 도움이 될 것이라 생각했다. 연휴 기간 동안에 에어 컴벳 게임을 제작하기로 결심했는데, 2학기 계획서에 설명한 바와 같이 충돌체와 파티클 효과 위주로 구현하고자 했다.

# 예상과 달리, 전투기의 자연스러운 무브먼트와 파티클 움직임을 구현하는데 많은 시간을 투자해야 했고 디테일한 구현에 어려움을 느끼며 기대에 미치지 못한 퀄리티를 만들게 되었다.

# 

# UE5에서 런타임 리매핑, 복잡한 입력처리 등을 수행할 수 있는 인풋 시스템 Enhanced Input을 사용해 전투기를 구현했다.

# Xbox 컨트롤러 입력에 매핑할 수 있도록 Input Mapping Context를 사용했다.

# BP Character 클래스는 CapsuleComponent를 이동 콜리전에 사용한다. (C++ 코드에서부터 protect 접근지정자로 사용된다) Character 만든 전투기의 Collision 형태를 변경하기 위해 Event BeginPlay 시 Capsule Component를 Simulate Physics로 set하도록 했다. (좋은 방법은 아니다. 수정한다면 BP Pawn 클래스로 처음부터 만드는 게 낫다.)

# DeltaSeconds 만큼 Roll, Pitch, Yaw한 수치를 현재수치와 타겟수치에 interp함수로 보간해 local Rotation으로 회전시킨다.

# (영상들)

# 

# 전투기 몸체에 맞춘 Collision을 사용해 충돌하는 모습.

# 

# 전투기 제트 분사가 CurrentSpeed에 따라 달라지는 모습.

# 나이아가라 커스텀 모듈을 생성해 원하는 방식으로 파티클 데이터를 처리하게 할 수 있다.

# 오른쪽 사진은 10개의 파티클이 인덱스를 가지며 나열된 모습으로 오른쪽에서 왼쪽으로 인덱스가 증가해 나간다.

# 인덱스가 N-1인 파티클의 위치를 Parent Position으로 두고 인덱스가 N인 Current Particle이 부모까지의 거리를 측정한다. 이때 거리가 미리 설정한 값보다 크다면 벡터를 계산해 추적해서 쫓아가는 방식이다.

# 즉, Array로 만든 파티클의 위치 값을 가져와 ribbon randerer를 통해 아래와 같은 움직임의 파티클 효과를 구현해 낼 수 있었다.

# 

# 

# 전투기 날개 소켓에 위 파티클 시스템을 부착시킨 지금까지의 결과물이다.

미 구현된 미사일 시스템과 투박한 HUD와 공허한 필드가 아쉽게 느껴지며, 특히 전투기 회전 시에 날개 플립도 그에 따라 움직이도록 만들고 싶었지만 실패했다.

스태틱 메시의 날개 플립이 있다면 가능했지만 기본 전투기 에셋이 하나 통짜였고, 다른 방법으로는 본 계층구조에서 소켓을 호출할 수 있었지만 그러기 위해서는 애님 그래프에서 작업하는 방식이 불필요하게 느껴져 처음 방식 그대로 제작했다.

나이아가라에 대해서 현재는 인덱스를 기준으로 파티클의 위치 값을 반환하는 정도에서 끝났지만, 나이아가라 외부의 데이터에 접근해 해당 데이터를 조작, 연산하는 법을 추가적으로 학습할 것이다. 유체 캐릭터 구현을 위해 나이아가라를 컴퓨트 셰이더로써 활용하게 될 것이다.

|  |  |  |  |
| --- | --- | --- | --- |
| **문제점 정리** |  | | |
| **해결방안** |  | | |
| **다음주차** | 16주차 | **다음기간** | 2023.10.10 ~ 2023.10.16 |
| **다음주 할일** | Skeletal Mesh나 RenderTarget에 접근하는 나이아가라 시뮬레이션 스테이지 실습 (조금 더 생각 필요) | | |
| **지도 교수**  **Comment** |  | | |