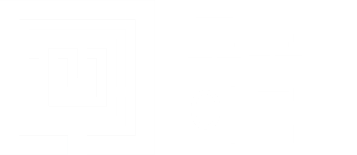
|  |
| --- |
| VR 프로그래밍 심화과정  with LiiYuu Senpai |
|  |
| 2021.09 ~ 2021.11  게임과 3학년  작성자: 함승호 |



# 1주차

# 강사소개, 수업계획, 입력매핑, 서버(이론)

# 2주차

# 냐옹

# 3주차

# 4주차

# 5주차

# 6주차

# 7주차

# 8주차

# 9주차

# 10주차

# 수업계획

# 여러가지 도구를 활용하는 네트워크 5대5 FPS 게임

# 패키징을 하여 배포까지 하는것이 목표.

# 참고게임 : RainbowSix Siege

# 

# 사용 버전 : UE4.27

# 입력 매핑

# 캐릭터 이동에 대한 Input 할당을 해줘야 한다.

# [세팅 → 프로젝트 세팅 → 입력]

# ( 경로 사진)

# FPS 게임이므로 이동에 WASD 를 사용할 것이다.

# 축매핑 옆의 + 를 누르면 새로운 축 매핑을 생성한다.

# (매핑 추가 사진)

# MoveForward 라는 이름의 축 매핑에 W, S 라는 키 값을 각각 W = 1, S = -1을 Scale 로 할당시킨다.

# 마찬가지로 MoveRight 축 매핑에 A = -1, D = 1 을 Scale 할당한다.

# x, Y축 및 번호 - 로열티 프리 그래프 스톡 사진

S

A

D

W

# 중앙에 캐릭터가 있다고 생각하면 Scale 할당이 이해 될 것이다.

# 캐릭터 C++ 클래스를 만든다.

# [파일→새로운C++클래스] 에서 부모클래스 Character 를 선택한다.

# (경로사진)

# (c++ 클래스 추가 사진)

# ' BaseCharacter ' 라는 퍼블릭 클래스를 만들었다.

# (완료사진)

# Visual Studio 에서 로드가 필요하다 할 것이다.

# 모두 로드 한다.

# Visual Studio를 꺼버렸다면 [파일 → Visual Studio열기] 로 다시 켤 수 있다.

# 솔루션 탐색기의 Games → Source → (프로젝트이름) → Public 에 에디터에서 생성한 BaseCharacter의 헤더 파일이 있다.

# class 프로젝트이름\_API ABaseCharacter : public ACharacter 클래스 내부에 설정한 입력 매핑들을 함수로 선언한다.

# (헤더파일 선언 사진)

# 파라미터인 float형 변수AixsValue 에는 매핑 설정시 Scale 에 넣어둔 값이 할당된다.

# ※팁 : 초록색 물결줄이 그어지는 이유는 선언만 하고 정의가 되어있지 않은것. 포인터를 변수명 우에 두고 ctrl + . 를 누르면 정의 만들기를 간단하게 할 수 있다.

# (cpp 정의 사진)

# Private 바이너리의 BaseCharacter.cpp 에 함수의 정의가 선언되었다.

# 이제 이동구현을 할 것이다.

# 정의한 함수 내부에 ~ 를 적는다.

# 그리고 클래스 내부에 ~ ("turn 이런거 ) 를 적는다

# 저장하고 에디터에서 컴파일시킨다.

# (컴파일 클릭 화면)

# 플레이 해서 이동을 확인한다.

# 이제 캐릭터의 시점을 회전시킨다.

# 마우스 회전에 대한 축 매핑을 할당한다.

# LookUp 축 매핑에는 키 값 마우스 Y , Scale = -1 로 할당한다.

# Turn 축 매핑에는 키 값 마우스 X , Scale = 1 로 할당한다.

# Yaw, Pitch, Roll 바로알기-스테이지 기초이론1

# Yaw :

# Networking and Multiplayer

# UE4 멀티플레이는 Client – Server 모델을 기반으로 한다.

# 즉 중앙에 서버가 있고, 클라이언트들이 주변에서 접근을 하는 방식이다.

# ( client to server)

# 서버의 구조는 2가지 존재한다.

# Listen Server : 클라이자 서버. → 서버의 역할도 하고 클라이언트의 역할도 하는 서버이다.(서버 상태로 플레이 가능)

# Dedicated Server : 서버의 역할만 하는 독립적인 서버. → 렌더링 관련 처리를 아예 하지 않는다. 별도 프로세스에서 서버 로직이 돈다.

# 소유권과 NetRole ○

# 네트워크 환경에서 로컬에 스폰되어 있는 모든 엑터들은 NetRole 을 가지고 있다.

# ※Netrole 에는 4가지의 종류가 있다.

# 1. Authority : 해당 엑터에 권한 O

# 2. Aturonomous Proxy : 권한 X, 소유권 O

# 3. Simulated Proxy : 권한 X, 소유권 X

# 4. None : 지정 Role이 없다.

# 서버에서 생성된 액터들은 기본으로 Authority 속성을 가지게 된다.

# 각 클라이언트들은 빙의한 폰에 대해 소유권을 가지게 된다.

# 서버에서 생성된 액터들은 냐옹이가 냐옹냐옹

# Has Authority 를 이용해서 엑터에 대한 권한을 가지고 있는지 알 수 있다.