

언리얼 프로그래밍 Part2-6

제목: 캐릭터 공격 판정

**강의 내용 : 충돌 판정 기능을 활용한 캐릭터의 공격 구현

**강의에서 다루는 게임 프레임워크 요소

강의에서 다루는 게임프레임워크 요소

게임	월드	모드	상태	
기믹	트리거	스폰	물리	
플레이어	입력	카메라	HUD	상태
폰	이동	모션	액션	위젯
데이터	애셋	테이블	설정	저장
인공지능	길찾기	BT		

이번 강의도 지난 강의와 동일하게

**강의 목표

강의 목표

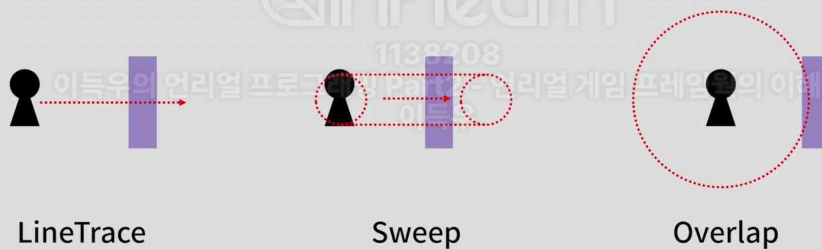
- 애니메이션 이벤트 발생을 위한 노티파이 설정 방법의 학습
- 공격 판정을 위한 트레이스 채널 설정과 판정을 시각적으로 디버깅하는 방법의 학습
- 대미지 프레임웍을 활용한 대미지 전달과 Dead 상태의 구현

노티파이의 설정 방법을 학습해보도록 하고요

***충돌 채널의 설정

캐릭터 액션의 충돌 판정

- 월드가 제공하는 충돌 판정 서비스를 사용
- 월드는 크게 세 가지의 충돌 판정 서비스를 제공함.
- 월드 내 배치된 충돌체와 충돌하는지 파악하고, 충돌한 액터 정보를 얻을 수 있음.



캐릭터가 수행하는 액션이 어떤 물체와 충돌하는지를 파악하는 방법은

- LineTrace : 지정한 방향으로 어떤 선을 투사하여 어떤 물체와 충돌되는지 감지하는 방법
- Sweep : 지정한 방향으로 어떤 도형을 지정한 방향으로 투사하여 어떤 물체와 충돌되는지 감지하는 방법 [LineTrace와 비슷]
- Overlap : 이 방법은 투사하는 것이 아니라 지정한 영역에 큰 범위의 도형을 설정하여 해당 볼륨 영역과 물체가 충돌되는지를 검사하는 방법

**트레이스 채널과 충돌 프로파일 생성

트레이스 채널과 충돌 프로파일 생성

- 액션 판정을 위한 트레이스 채널의 생성 : ABAction. 기본 반응은 무시
- 캐릭터 캡슐용 프로파일 : ABAction 트레이스 채널에 반응. 오브젝트 타입은 Pawn
- 스케레탈 메시용 프로파일 : 랙돌 구현을 위해 주로 활용됨.
- 기믹 트리거용 프로파일 : 폰 캡슐에만 반응하도록 설정. 오브젝트 타입은 WorldStatic



-이러한 충돌 판정 기능 구현을 위해 UE5가 제공하는 물리 설정에 프로젝트 상황에 맞춰서 채널이나 프로파일을 설정해주는 것이 좋다.

-먼저 우리가 제작한 ABCharacter에 액션판정을 위한 트레이스 채널을 생성해볼 것이다.(ABAction), 기본 설정은 무시로 설정하여 다른 물체와 반응이 일어나지 않도록 설정할 것이다.

-그리고 나서 우리가 제작한 캐릭터와 앞으로 제작할 기믹에 물리적인 충돌을 관리할 수 있도록 충돌 프로파일을 프로젝트에 맞게 생성해주면 좋다

-캐릭터의 경우에는 두 가지를 고려해야 하는데 루트 컴포넌트인 캡슐 컴포넌트의 프로파일과 스케레탈 메시의 스케레탈 메시용 프로파일이다.

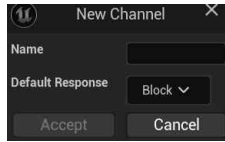
-캐릭터의 루트컴포넌트인 캡슐 컴포넌트의 경우에는 우리가 생성한 트레이스 채널과 반응해야 하기에 캡슐 컴포넌트를 위한 충돌 프로파일을 별도로 생성해야한다. 특히 ObjectType을 Pawn으로 설정해야 한다.[이를 우리가 새롭게 ABPawn이라는 이름으로 지을 수도 있지만 우선 엔진이 기본으로 제공하는 Pawn으로 설정하자]

-스케레탈 메시용 프로파일은 주변의 사물과 흐느적거리면서 반응하는 랙돌(Ragdoll)같은 기능을 구현할 때 주로 활용이 된다. 우리가 이번에 만든 프로젝트는 이것을 사용하지 않기에 NoCollision 프로파일을 적용한다.

-그 다음으로 기믹 안에 설치된 트리거용 프로파일이다. 트리거용 프로파일은 우리가 오브젝트로 지정한 폰에 대해서만 반응하고 나머지는 모두 다 무시로 설정하도록 새로운 프로파일을 생성하도록 한다.

-기믹의 경우 움직이지 않기 때문에 objectType을 World Static으로 선언한다.

-NewTraceChannel



기본 반응을 Block이라고 설정하면 해당 트레이스로 어떤 충돌 감지 액션을 구현할 때 모든 물체가 반응하게 된다. 우리는 우리가 지정한 캡슐 컴포넌트만 반응하도록 할 것이기에 Ignore로 설정한다.

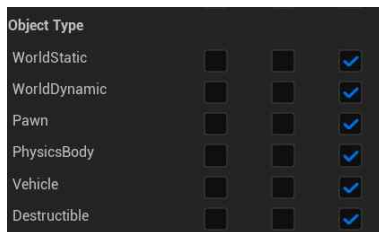
-TraceType



Camera 채널의 경우 지난 강의에서 설명한 카메라를 당겨주는 카메라 콜리전 테스트에 사용하는 트레이스의 채널이다. 따라서 만약 월드에 배치한 다른 캐릭터가 내 캐릭터의 시야를 방해한다면 이 카메라 채널에 의해 블록 반응이 일어나 앞으로 당겨진다.

[TraceType의 경우에는 Ignore와 Block만 신경써주면 된다.]

-ObjectType



오브젝트 타입의 경우 Ignore와 Overlap, block 3가지를 고려해야 한다. Overlap은 길을 막지는 않지만, 이벤트가 발생하도록 설정하며, block은 길을 막는다.

■ 가정 : Destructible을 사용하여 월드에 배치한 물체를 파괴했다고 가정해본다

이 파괴한 조각들이 길을 막으면 불편해지는데 이런 경우 보통 Ignore로 설정하는 것이 좋을 수가 있다.[필요에 따라 우리가 설정한다] 우리프로젝트는 사용 안할 예정임.

-설정 경로

설정을 완료하면 E:\HighProject\ArenaBattle\Config에 있는 DefaultEngine.ini파일에 바로 저장된다. 확인하면 profile내용이 많이 써져 있는데 이는 우리가 엔진에서 기본적으로 제공하는 정보를 수정하였기 때문에 다시 새로운 정보를 덮어 씌운다라고 보면 된다.

■ DefaultChannelResponses라는 정보의 ABAction이란 이름이 엔진에서 지정하고 있는 내부 열거형의 ECC_GameTraceChannel1라는 열거형 값을 사용한다고 지정한다 이는 ABAction이라는 이름을 지정했지만 코드 내에서는 ECC_GameTraceChannel1를 사용하는 것이 좀 더 편리하다. [해당 정보를 사용하여 우리가 공격판정 기능을 구현해보도록 할 것이다.

-몽타주에서 이벤트 프레임을 정할 때 이벤트를 발생시킬 때 좋은 것이 Notify 기능이다.

***공격 판정의 구현

**월드 트레이싱 함수의 선택

월드 트레이싱 함수의 선택

- 세 가지 카테고리로 원하는 함수 이름을 얻을 수 있음.
- 카테고리 1 : 처리 방법
- 카테고리 2 : 대상
 - Test : 무언가 감지되었는지를 테스트
 - Single 또는 AnyTest : 감지된 단일 물체 정보를 반환
 - Multi : 감지된 모든 물체 정보를 배열로 반환
- 카테고리 3 : 처리 설정
 - ByChannel : 채널 정보를 사용해 감지
 - ByObjectType : 물체에 지정된 물리 타입 정보를 사용해 감지
 - ByProfile : 프로파일 정보를 사용해 감지

{처리방법}{대상}{처리설정}

월드에서 제공하고 있는 트레이싱 함수들이 있는데요

-첫 번째 카테고리는 처리 방법으로 : Linetrace/Sweep/Overlap 이 세가지중 하나를 접두사로 붙여주면 된다.

-두 번째 카테고리

- 무언가 감지 되었는지 테스트하는 용도로 사용하고 싶으면 TEST로 넣자.
- 하나에 대해서 내가 감지된 액터의 정보를 얻고싶을 땐 Single 또는 AnyTest라는 단어 붙이자.[Linetrace/Sweep은 Single, Overlap은 AnyTest를 붙여주면 된다.]
- 영역내에 있는 감지된 모든 정보를 얻고 싶으면 Multi를 사용해준다.

-세 번째 카테고리

- 우리가 지정한 트레이스 채널을 사용할거라면 'ByChannel'
- 지정된 오브젝트 타입을 사용할거라면 'ByObjectType'
- 물리에 설정된 프로파일 정보를 그대로 사용할거라면 'yProfile'

**캐릭터 공격 판정의 구현

캐릭터 공격 판정의 구현

- 캐릭터의 위치에서 시선 방향으로 물체가 있는지 감지
- 작은 구체를 제작하고 시선 방향으로 특정 거리까지만 투사.
- 하나의 물체만 감지
- 트레이스 채널을 사용해 감지

Inflearn

1138208

이득우의 언리얼 프 {처리방법} {대상} {처리설정} 레임웍의 이해

이득우



{Sweep}{Single}{ByChannel}

우리가 캐릭터를 공격할 때에는

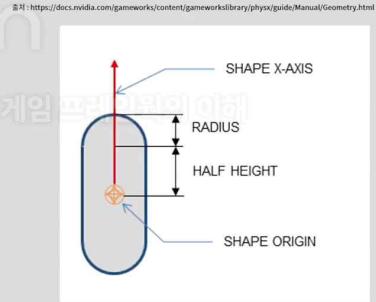
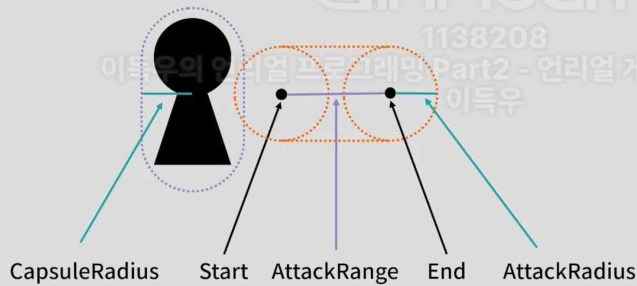
-우리는 캐릭터의 위치에서 시선 방향으로 물체가 있는지를 투사해서 감지할 예정이기에 LinTrace/Sweep 둘중에 하나를 사용해야 하는데 우리는 작은 구체를 제작하고 시선 방향으로 투사할 것이기에 Sweep을 사용해보도록 하자

-기능을 구현하더라도 우리가 이것이 감지 됐는지 알아보는 방법은 로그를 찍는것밖에 없다 하지만 로그를 찍는게 아니라 감지된 영역을 직접 화면에 보여주면 더 좋다. 따라서 우리는 언리얼 엔진이 제공하는 드로잉 디버깅 기능을 사용하여 우리가 트레이싱을 지정한 영역이 물체에 감지 되었는지를 한번 표시해보겠다.

**물체 충돌 테스트

물리 충돌 테스트

- 디버그 드로잉 함수를 사용해 물리 충돌을 시각적으로 테스트
- 90도로 회전시킨 캡슐을 그리기
 - Origin
 - HalfHeight
 - Radius



-충돌 테스트를 위해서 트레이싱한 영역은 위 그림과 같이 설정이 된다.

■ 시작 지점과 끝지점 그리고 우리가 지정한 어택의 길이와 구체에 대한 반지름으로 구성됨.

-캐릭터 자체에도 캡슐이 있고, 캡슐의 반지름이 위 그림처럼 지정이 되어 있다. 이러한 정보를 바탕으로 우리가 투사한 전체 영역을 한번 그려볼 것인데 캡슐 모양을 가지고 있다.

-캡슐들은 위 그림에서 우측 하단에 나오듯 다양한 용어들이 있다.(aduis, Half Height, Shape Origin)

■ 캡슐의 경우 기준점이 있다.

■ 다음에 둥근 부분을 제외한 전체 길이의 절반값을 Half Height라고 한다.

■ 캡슐에 사용되는 구체의 반지름을 Radius라고 한다.

■ 다음 앞서 설명한 3가지로 캡슐 영역을 만들 수가 있는데 트레이싱한 영역은 넓혀져 있는 상태이기 때문에 회전을 가해주면 우리가 위 그림처럼 넓힌 캡슐을 직접 디버그 드로우를 사용하여 그려줄 수 있다.

**정리

트레이스 채널을 활용한 공격 판정

1. 공격 판정 구현을 위한 물리 트레이스 채널 및 프로필 설정
2. 디버그 드로잉 기능을 활용한 충돌 디버깅
3. 대미지 프레임워크를 사용한 대미지 전달
4. 델리게이트와 람다함수의 간편한 활용

공격 판정 기능을 구현해 보았는데요