15장 총과 슈터

인터페이스 : 인터페이스란 외부와 통신하는 공개통로이며 메서드 이름 제공자

인터페이스는 선언만 있고 구현이 없기 때문에 인터페이스를 상속받아 사용하는 사용자들은 무조건 구현을 해야함.

이렇게 하는 이유는 추상화를 통해 오브젝트의 Use메소드를 다양하게 구현할수 있고 if문의 사용을 줄여 자원관리에 효과적이다.

인터페이스를 상속한 클래스는 메서드를 반드시 public으로 구현해야한다.

인터페이스를 사용함으로써 느슨한 커플링구현 가능

총 오브젝트 준비

총과 슈터를 분리하여 구현, 재활용이 쉽다.

총 구현: 빈 엠피트를 캐릭터 자식으로 넣고 gun pivot으로 이름변경 오른쪽 팔꿈치에 존재할수있도록 위치조정 추후 OnAnimatorIK()를 통해 항상 오른쪽 팔꿈치에 고정시킬 예정

라인렌더러 생성 :주어진 점들을 이은 선을 그리는 컴포넌트 gun에 추가

총알 궤적을 구현하기위한 것과 총을 쐈을시 적이 맞았는지 아닌지를 구현할 때 사용

총구위치와 탄알이 닿을 위치 2가지를 할당함

상태표현타입 enum 열거형 상수를 이용해서 구현

Public enum State{

Ready,

Empty,

Reloading //위에서부터 0.1.2번을 할당받아 사용가능1

}

코루틴

대기시작을 가질수 있도록 하는 메서드로 조건이 만족되야 밑의 코드를 실행하도록 할수있음

또는 그냥 쉬기도 가능

접근제한자 IEnumerator 메서드이름(){

} 형태로 구현



레이캐스트

레이캐스트는 보이지 않는 광선을 쐈을 때 광선이 다른 콜라이더와 충돌하는지 검사하는 처리.

Ray라는 타입으로 사용가능

레이캐스트를 실행했을 때 레이가 콜라이더를 가진 오브젝트와 충돌하면 ,raycastHit타입의 충돌 정보가 생성, 출동한 오브젝트 출동 위치 충돌한 표면의 방향들을 알수있음

Physics.Raycast() 메서드는 충돌 정보를 반환해준다. Raycast()메소드는 단순히 충돌여부를 bool값으로 전달해주기 때문에 out키워드에 hit정보를 담아서 내보낸다.

슈터 만들기

IK이해하기

IK를 시작하기 전에 먼저 FK를 알아야 한다. 캐릭터 애니메이션은 기본적으로 FK(전진 운동학)로 동작함

FK에서는 부모 조인트에서 자식 조인트 순서로 움직임을 적용한다. 부모 조인트가 움직이면 자식 조인트는 자동으로 움직임

FK는 쿤단위의 관절에서 세부적인 관적 순서로 움직임을 적용해서 적용된 최종 결마물, 손의 위치를 먼저 정하고 애니메이션을 변경할수 없으므로 물건을 손의 위치로 이동시켜야함.

IK는 자식의 조인트 위치를 먼저 결정하고 부모 조인트가 거기에 맞춰 변형되는 것.

IK의 각 부분에 맞춰서 위치 웨이트, 회전 값등을 설정해준다.