10장 공간과 움직임

공간과 움직임

월드는 하나이지만 여러가지의 좌표계를 사용하여 표현

전역 공간, 오브젝트 공간 ,자식공간 등등

전역공간 : 월드의 중심이라는 절대 기준이 존재하는 공간

피벗과 센터 피벗은 오브젝트의 실제 기준점을 기준, 센터는 눈으로 보이는중점 기준으로 오브젝트를 배치

오브젝트 공간 :오브젝트를 생성하고 y좌표를 90을 꺽으면 y축이 z축과 일치하게 되므로 z값을 변경하여도 z값이 변경되는것이 아니라

x값이 변동된다. 오브젝트가 바라보는 시점과, 전역좌표계가 인지하는 값이 다름을 명심하자

오브젝트의 부모 자식 관계 : 부모 객체가 움직여지면 자식 객체도 같이 움직여진다.

상속을 받게 되면 지역공간을 만들어서 부모 객체를 중심으로 사물을 보게되기 때문이다.

부모 객체가 존재하지 않을 경우 전역공간의 원점(0,0,0)을 기준으로 좌표를 측정

전역공간,지역 공간 ,오브젝트 공간

전역공간: 게임월드의 원점을 기준으로 위치를 측정

지역공간 : 자신의 부모 게임 오브제그를 기준으로 위치를 측정

오브젝트 공간 : 자기 자신을 기준으로 위치를 측정

지역공간에서 위치, 회전,스케일값 측정 : 부모 게임오브젝트를 기준으로 측정

지역공간에서 평행이동 : 게임 오브젝트 자신의 방향을 기준으로 평행이동

transform.Translate(new Vector3(0, 1, 0,Space.World)) 은 전역활동

을 transform.position = transform.position + tranfrom.up\*1);로 표현할수있다

transform.Translate(new Vector3(0,1,0,Space.Sleft)은 지역활동 기본은 지역활동이다

transform.Rotate()도 마찬가지로 전역과 지역을 가짐

이때 transform.Rotate(new vector(0,60,0))처럼 사용할수있는데

Vector rotation이 아니라 이미 변환된 Quanerion 에 벡터값을 넣는거라서 허용된다고 생각한다.

Vector 속기방법

종류

transform.forward : new Vector(0,0,1));

transform.back : new Vector(0,0,-1));

transform.right : new Vector(1,0,0));

transform.left : new Vector(-1,0,0));

transform.up : new Vector(0,1,0));

transform.down : new Vector(0,-1,0));

f