6강 몬스터 ai 구현하기

6-1 준비하기

적절한 맵타일을 그려주고 에너미스프라이트 추가하고 idle run 애니메이션 클립 상태 트랜지션 파라미터 전부 설정해준다.

6-2 기본이동

//물리기반으로 움직이니 rigidbody필요

Rigidbody2D rigid;

void FixedUpdate()

{

rigid.velocity = new Vector2(x축 속도값, rigid.velocity.y);

}

6-3 행동 결정 로직

//행동설정 로직 3가지 왼쪽이동, 정지, 오른쪽이동

public int nextMove;//행동지표를 결정할 변수를 설정했다.

//-1 왼쪽, 0 정지, +1 오른쪽

void Awake()

{

rigid = GetComponent<Rigidbody2D>();

Think(); //시작할때 랜덤으로nextMove에 값이 설정된다.

}

void Think()

{

nextMove = Random.Range(-1, 2);//최대값은 랜덤값에 포함이 안되므로 2로 설정

Invoke("Think", 3);

//함수가 자기자신을 호출하는 재귀로 만들어준다. 근데 이러면 너무 많이 호출하므로 invoke는 필수다.

//Awake()에서 Think()하고 Think는 일정시간뒤에 다시 Think로 nextMove를 넣어줌.

}

void FixedUpdate()

{

rigid.velocity = new Vector2(nextMove, rigid.velocity.y);

}

6-4 지능 높이기

바닥에 빠지는 문제가 있다. 이것을 해결하기 위한 방법 “지형 체크” = 레이캐스트

void FixedUpdate()

{

//기본이동

rigid.velocity = new Vector2(nextMove, rigid.velocity.y);

//platform check 몬스터가 한수앞을 바라보고 자기 바로 밑에가 아니라

//자기 가는 방향의 한칸 앞이 낭떠러지인지 체크해야함

Vector2 frontVec = new Vector2(rigid.position.x + nextMove\*0.5f, rigid.position.y);

//새로 정의하는 앞쪽 벡터를 자세히보면 에너미의 x포지션에서 nextMove를 더한 좌표값을 가진다. 이는

//왼쪽을 바라보면 -1이고, 오른쪽을 바라보면 +1이고 가만히 있는 상태면 0을 더한다.

Debug.DrawRay(frontVec, Vector3.down, new Color(0, 1, 0));

//에디터 상에서만 ray를 그려주는 함수이다. 시작위치는 에너미의 위치가 아니라 에너미의 진행방향 한칸 앞이 될것이다.

RaycastHit2D rayHit = Physics2D.Raycast(frontVec, Vector3.down, 1, LayerMask.GetMask("Platform"));

//레이캐스트2d를 담을 변수를 선언하고 physics2d.raycast()함수를 이용하여 생성한다.

//Physics2D.Raycast(레이캐스트 시작점, 방향, 크기);

//레이캐스트함수를 이용해 쏜 정보가 rayhit에 담긴 형태이다.

//마지막에 있는 레이어마스크를 설정하면 그거에 해당하는 콜라이더만 담을것이다.

if (rayHit.collider == null) //충돌된 콜라이더를 검사해서 null이라면 즉 낭떠러지라면

{

nextMove \*= -1; //nextMove 부호 반전

CancelInvoke();// 이 함수를 쓰면 지금 초시계가 돌아가고 있는 Invoke()를 멈춘다.

Invoke("Think", 3); //이거할려고 위에 인보크 취소한거

}

}

6-5 애니메이션 설정

기존에 애니메이션 컨트롤은 setBool 값을 이용해 조건을 만족하면 트랜지션으로 상태를 바꾸는 것을 구현했으면 이번엔 int값 체크로 해결한다.

//애니메이션 컨트롤

animator.SetInteger("runSpeed", nextMove);

//방향 전환

if(nextMove != 0)

spriteRenderer.flipX = nextMove == 1;