2d 게임 만들어보기

2d의 특징 이미지 파일을 스프라이트 타입으로 임포트

카메라가 직교모드

라이팅 설정 제한

씬창이 2d

스타트지점 만들기 및 데드존 만들기

스타트지점은 기본생성, 데드존은 태그를 dead로 하고 트리거 만들기

스프라이트란 단일 이미지파일에 캐릭터의 여러모습을 합친 스프라이트 시트

리지드바디로 중력을 적용시켜주고 ,z축은 고정시켜서 캐릭터가 회전하지 않게 해준다.

써클콜라이더컴포넌트를 해서 발판위에 물리적 충돌을 할수있도록 해줌

리지드 바디 3 컴포넌트 충돌 감지 방식은 이산과 연속 두가지가 있음

이산은 충돌 감지를 일정시간 간격으로 끊어서 실행.

연속은 이전위치, 충돌까지 감지

오디오소스는 오디오 컴포넌트 추가후 소스파일을 선택해준다.

애니메이션

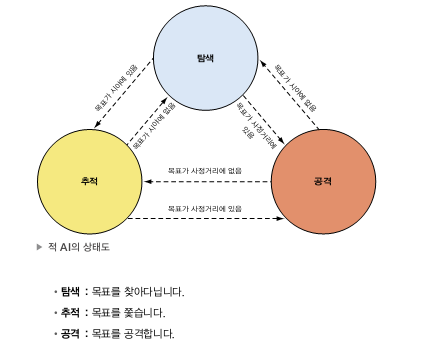
윈도우에서 애니메이션 창 열기 선택, 해당하는 스프라이트 파일 선택후 전체 집어넣기

오른쪽 톱니바퀴를눌러서 설정 프레임 변경

달리기와,점프, 죽음 애니메이션 생성

유한상태 머신

유한상태 머신은 유한한 수의 상태가 존재하며, 한 번에 한 상태만 현재 상태가 되도록 프로그램을 설계하는 모델



예를들어 탐색,추적,공격이 세가지로 AI가 구성되있을시 한가지 상태만을 가지며 행동을 취함

진행중인 프로젝트에서는 런,점프,다이가있으며

기본포함상태에는 엔트리 종료 any state가 있음

anystate는 조건만 갖춰진다면 현재 상태가 무엇이든지 전이를 가능하게 만듬

전이 설정하기 지정된 애니메이션을 누르고 오른쪽 마우스로 전이연결가능

파라미터값에서 조건들 지정가는, has exit time은 조건을 만족해도 행동이 끝나기전까진

안나가는 지는것 캐릭터들 스킬 딜레이로 보면될듯

Transition duration 애니메이션의 동작과 동작을 연결하여 자연스럽게 보여지도록 하는것이지만

2d에서는 지연시간만 발생시키므로 사용을 거의안함

애니메이터의 파라미터 사용가능 종류 float , int , bool , trigger가 있음

플레이어 스크립트 작성하기 :

제일먼저 사용할 변수들과 컴포넌트 선언

그다음 스타트메서드속에 사용할 컴포넌트들을 가져와 변수에 할당

변수이름 = GetComponent<컴포넌트이름>(); 제너릭함수를 사용해야 구조체처럼 가져올수있다.

업데이트 메서드에는 프레임마다 실행된 이벤트들을 선언

죽었을때

마우스 왼쪽버튼을 누르거나 점프수가 2이하일때 이거나 떼는순간 && 위로상승중이라면 속도조정하는 메소드

현재 땅에 닿아있는지 아닌지를 상태를 변환

이렇게 세가지 작성됨

죽은상태가 트루이면 돌려보내서 업데이트 밑에부분이 실행 안되게함

마우스클릭과 점프수 이하일때는

점프수 카운트를 늘리고 jumpCount++;

잠시 멈추게함 playerRigibody.velocity = Vector2.zero;

캐릭터를 위로 이동시킴(점프) playerRegibody.AddForce(new Vector2(0,jumpForce));

오디오 재생 playerAudio.Play();

input.getmouseButtonDown()은 매개변수로 int값을 받아 어떤 입력을 받았는지 인식하고

0은 좌클릭 1은 우클릭 3은 휠클릭이다

input.getmousebuttonDown() : 마우스 버틀을 누르는순간

input.getmousebutton() : 마우스 버틀을 누르고 있는동안

input.getmousebuttonUp() : 마우스 버튼을 때는 순간

잠시 멈추게 하는 이유는 멈추지 않고 버튼을 빠르게 두번클릭하는 경우

첫번째 점프와 두번째 점프의 힘이 합쳐서 상대적으로 더 높게 뛰게된다.

grounded값을 바꾸는 방법은 start에서 불러온 animator를 이용

animator.SetBool("Grounded",isGrounded); // 전환조건에 grounded를 했기때문에 애니메이션과 함께 바뀜

Die()메서드

animator.setTrigger("Die") 로 die로 전환시킴;

playerAudio.clip = 클립이름

playerAudio.Play();

playerRigidbody.velocity = Vector2.zero;

isDead = true;

OnTriggerEnter2D 메서드

충돌한 대상의 태그를 확인하고 Dead이며 플레이어가 죽지 않았다면 사망처리한다

private void OnTriggerEnter2D(Collider2D other) {

if (other.tag == "Dead" && !isDead)

{

// 충돌한 상대방의 태그가 Dead이며 아직 사망하지 않았다면 Die() 실행

Die();

}

}

OnTriggerEnter2D(Collider2D other)와 OnCollisionEnter2D(Collision2D collision)에서 차이는

트리거 물질과 닿았는지 아닌지 차이와 매개변수로 other은 충돌한 대상(트리거는 상대방 값만 제공),

collider는 물체컴포넌트부모 클래스이고, collision은 충돌정보이다

onTrggierEnter는 monobehaviour의 자식으로 오브젝트에 연결된 트리거안에 다른 오브젝트가 들어갔을때 호출

collision에 담겨있는 충돌을 contacts[]배열에 정보로 저장하고 normal은 노말 벡터 y값이 >0.7 이상이면 위쪽을 바라고보고 있는 형태이다.

private void OnCollisionEnter2D(Collision2D collision) {

// 어떤 콜라이더와 닿았으며, 충돌 표면이 위쪽을 보고 있으면

if (collision.contacts[0].normal.y > 0.7f)

{

// isGrounded를 true로 변경하고, 누적 점프 횟수를 0으로 리셋

isGrounded = true;

jumpCount = 0;

}

}

private void OnCollisionExit2D(Collision2D collision) {

// 어떤 콜라이더에서 떼어진 경우 isGrounded를 false로 변경

isGrounded = false;

}