

Unity 포트폴리오

생존 마인크래프트

장르 : 생존 어드벤처

김 준

유튜브 영상 링크 : <https://youtu.be/m1vydtjn4RY>

목 차

1. 플레이어(애니메이션)	3
2. 인벤토리	4
3. 아이템 제작	6
4. 에너지	7
5. 아이템	8
6. 오브젝트	9
7. 무기	10
8. 날씨	11
9. 커스터마이징	12

1. 플레이어

1) 플레이어

-첫 상태는 IDle 상태로 시작합니다. 플레이어는 행동 중 이동이 불가능하게 Bool 값을 사용해 검사하였습니다.

2) 애니메이션 변경

-오브젝트에는 실제 콜리더와 인식용 콜리더가 존재합니다. 인식 콜리더 안에 들어온다면 플레이어의 애니메이션이 변경하도록 하였습니다.

플레이어의 애니메이션을 구분하기 위해 int 를 사용하여 구분하였습니다.

<애니메이션 구분을 쉽게 하기 위하여 인트를 사용>

3) 플레이어 스텟

-플레이어 스텟은 싱글톤으로 만들어 접근을 쉽게 하였습니다.

생존게임이다 보니 이곳저곳에서 플레이어의 스텟에 관한 정보 값이 필요해 싱글톤으로 하였습니다.

2. 인벤토리

1) 데이터 베이스

-아이템은 많은 곳에서 필요로 하기 때문에 싱글톤으로 만들어 접근이 쉽게 만들었습니다 .

아이템의 이름, 아이콘, 설명, 능력치 등을 선 입력후 List 에 저장해 보관을 하는 방식으로 만들었습니다.

2) 아이템 슬롯

List 의 인덱스에 해당하는 아이템의 정보를 가져와 복사합니다.

슬롯은 슬롯보다 상위인 인벤토리라는 코드에서 한 번에 관리를 합니다.

3) 인벤토리 코드

-배열에 아이템 슬롯을 넣어 가지고 있습니다.

아이템을 넣을 때 순차적으로 넣을 때 활용하거나 인벤토리를 검사할 때 사용합니다.

4) 아이템 이동

-마우스로 클릭을 하면 클릭된 대상의 아이템 정보 값의 여부를 확인 후 존재한다면 클릭했던 슬롯과 아이템의 정보를 마우스를 따라다니는 슬롯으로 이동시킵니다. 만약 아이템을 짚은 상태에서 마우스 키를 놓으면 처음에 저장된 위치로 다시 이동합니다. 아이템을 다른 슬롯에 놓을 때 먼저 타겟 슬롯에 아이템이 있는지 없는지 확인을 합니다. 있다면 처음 클릭했던 아이템과 위치가 바뀌게 되고 없다면 마우스 슬롯에 있는 아이템의 정보가 타겟 아이템 슬롯의 정보로 이동하게 됩니다. 만약 타겟이 쓰레기통이라면 마우스 슬롯의 정보를 삭제시키게 됩니다.

3. 아이템 제작

-아이템 제작칸에 아이템을 드랍시키면 먼저 List 에 드랍된 아이템의 인덱스를 저장합니다.

이렇게 저장된 정보를 제작 버튼을 누르면 아이템이 생성되며 동시에 List 와 아이템 제작칸의 아이템들을 삭제시키고 완성칸에 아이템을 생성합니다. 이 아이템 또한 아이템 데이터베이스의 값을 가져와 사용합니다.

4. 에너미

유니티 Navigation 을 사용하였습니다.

1) 박쥐

-박쥐에겐 타격을 맞는 콜리더와 인식용 콜리더 2 가지가 존재합니다. 인식용 콜리더는 트리거 판정으로 물리적 충돌을 무시하게 하였습니다.

인식용 콜리더 안에 플레이어가 충돌한다면 플레이어에게 다가가 공격하는 형식입니다.

에너미를 죽이면 에너미마다 각각 다른 아이템을 인벤토리로 넣게 됩니다.

2) 유령

-유령도 2 개의 콜리더를 가지고 있습니다.

유령은 공격 모션이 없고 몸 콜리더와 충돌시 데미지를 입는 형식으로 하였습니다.

피격중에는 공격과 이동이 불가능 하게 만들었 습니다. 사망시 사망 이펙트와 사운드를 출력후 삭제됩니다.

에너미는 사망시 StartCoroutine 함수를 호출해 일정 시간이 지나면 초기화된 상태로 부활합니다.

5. 아이템

-아이템은 제작용 아이템과 착용용 아이템 또 식용용 아이템으로 크게 3개로 구분하였습니다.

1) 제작용 아이템

-어떠한 곳에도 사용이 불가능하며 제작 이외에는 사용이 불가능하도록 만들었습니다.

2) 착용용 아이템

-인벤토리 장비 창에서 착용 시 능력치가 상승하게 만들었고 착용 이외는 사용이 불가능합니다.

3) 식용용 아이템

-제작과 식용이 가능합니다. 아이템 제작 시 사라지며 새로운 아이템으로 탄생하게 되고 식용 시 아이템이 사라지며 저장된 아이템의 정보만큼의 능력치가 상승하게 됩니다.

6. 오브젝트

-오브젝트는 주로 트리거 충돌을 사용했습니다.

1) 획득형

-플레이어가 오브젝트와 충돌 시 지정된 텍스트가 출력되고 키를 입력시 아이템은 사라짐과 동시 인벤에 아이템이 들어오게 됩니다.

2) 농사형

-충돌 시 지정된 텍스트가 출력됩니다.

키 입력시 플레이어가 땅에 씨앗을 심습니다.

씨앗은 시간이 지나면 성장해 농작물이 됩니다.

이 농작물 위에서 키 입력시 인벤토리에 들어오게 됩니다. 농작물을 획득 후 일정 시간 동안 씨앗을 심을 수 없게 됩니다.

3) 가구형

-키를 입력받으면 아이템을 bool 값을 통해 인벤토리에 푸쉬하게 됩니다.

7.무기

1)근거리

-무기에는 충돌용 콜리더를 부착시켰습니다.

공격을 실행하면 콜리더를 켜주고 충돌하거나 공격 애니메이션이 종료되면 다시 콜리더를 꺼주는 식으로 하였습니다.

2)활

-발사전 인벤토리에 화살이 존재하는지를 검사합니다. 검사는 인벤토리 코드를 통해합니다. <목차 인벤토리 참조>

존재한다면 아이템의 개수를 감소시키며 화살을 발사하도록 하였습니다.

8.날씨

-게임시간 1 시간을 현실 시간 1 분으로 잡았습니다.

태양은 게임시간 1 시간마다 15 도씩 이동하게 됩니다 또 플레이어의 능력치를 감소시킵니다.

오전과 오후에 따라 감소대는 능력치가 다르도록 설정했습니다.

1 시간씩 지날 때마다 랜덤을 돌아 20% 확률로 흐린 날씨가 됩니다.

흐린 날씨가 되면 능력치 감소가 다르게 적용됩니다.

9.커스터마이징

-UI 는 UGUI 를 사용하였습니다(NGUI 사용 가능)

처음 모든 캐릭터의 SetActive 를 False 로 만들고 시작합니다.

캐릭터 UI 를 클릭하면 해당 SetActive 를 True 로 만들며 랜더 시킵니다.

회전, 색, 얼굴 버튼은 인트 형으로 구분하였습니다.

게임 플레이 버튼을 누르면 캐릭터의 정보와 색 얼굴의 인덱스를 게임씬으로 넘겨주고 플레이어는 그것을 받아 색과 얼굴을 바꾸어 줍니다.