

Week6

B24 퀘스트 시스템 구현하기

1. 퀘스트 대화

Quest 매니저 생성. QuestData 생성

questData는 구조체로 사용. Dictionary 사용.

Npc id를 받고, 퀘스트 번호를 반환하는 함수 생성

2. 퀘스트 진행

퀘스트번호+ npc id = 퀘스트 대화 데이터 id

Talkdata에 추가한다. 근데 그냥하면 순서 상관없이 대화가 진행된다.

퀘스트매니저에서 퀘스트 데이터 만들때의 그 순서를 지켜야한다. -> 대화진행을 위해 퀘스트 인덱스를 올린다. -> 함수생성.

```
//set talk data
int questTalkIndex = questManager.GetQuestTalkIndex(id);
string talkData = talkManager.GetTalk(id + questTalkIndex, talkIndex);
```

순서에 맞게 대화했을 때만 퀘스트 대화순서 올림. 다음 퀘스트를 위한 함수도 생성

Talk를 받아올 때 npcid + 퀘스트 번호를 받아온다.

3. 퀘스트 오브젝트

코인을 새로 만들어주고, questmanger에 gameobject 만들어준다.

퀘스트 오브젝트를 관리할 함수를 생성한다. 퀘스트 번호, 퀘스트 대화 순서에 따라 오브젝트를 조절한다.

4.예외처리

퀘스트가 없을 때! -> %를 사용하여 기본대사 또는 처음 대사를 가지고 온다. 이때는 containskey를 사용한다.

```

if (!talkData.ContainsKey(talkIndex))
{
    //해당 퀘스트 진행 중 대사가 없을 때
    //퀘스트 맨처음 대사를 가지고 온다.

    if(!talkData.ContainsKey(id - id % 10))
    {
        //퀘스트 맨 처음 대사마저 없을 때
        //기본 대사를 가지고 온다.
        if (talkIndex == talkData[id].Length)
            return null;
        else
            return talkData[id - id % 100][talkIndex];
    }
    else
    {
        if (talkIndex == talkData[id].Length)
            return null;
        else
            return talkData[id - id % 10][talkIndex];
    }
}

```

5. 로직 다듬기

반환 값이 있는 재귀함수는 return까지 써줘야한다.

오버로딩. 매개변수로 같은이름의 함수 구분

questManager

```

@ Unity 스크립트 | 참조 1개
public class QuestManager : MonoBehaviour
{
    public int questId;
    public int questActionIndex;
    public GameObject[] questObject;

    Dictionary<int, QuestData> questList;

    @ Unity 메시지 | 참조 0개
    private void Awake()
    {
        questList = new Dictionary<int, QuestData>();
        questObject[0].SetActive(false);
        questActionIndex = 0;
        GenerateData();
    }

    참조 1개
    void GenerateData()
    {
        questList.Add(10, new QuestData("마을 사람들과 대화하기", new int[] { 1000, 2000 }));
        questList.Add(20, new QuestData("루도의 동전 찾아주기", new int[] { 5000, 2000 }));
        questList.Add(30, new QuestData("퀘스트 클리어", new int[] { 0 }));
    }

    참조 1개
    public int GetQuestTalkIndex(int id)
    {
        return questId + questActionIndex;
    }

    참조 1개
    public string CheckQuest(int id)
    {
        //new talk target
        if(id == questList[questId].npcId[questActionIndex] )
            questActionIndex++;

        //control quest object
        ControlObject();

        //talk complete && new quest
        if (questActionIndex == questList[questId].npcId.Length)
            NextQuest();

        //quest name
        return questList[questId].questName;
    }

    참조 1개
    void NextQuest()
    {
        questId += 10;
        questActionIndex = 0;
    }

    참조 1개
    void ControlObject()
    {
        switch (questId)
        {
            case 10:
                if (questActionIndex == 2)
                    questObject[0].SetActive(true);
                break;
            case 20:
                if (questActionIndex == 1)
                    questObject[0].SetActive(false);
                break;
        }
    }
}
```

Talk manager

```
void GenerateData()
{
    //talk data
    //npc a : 1000, b : 2000, desk 3000, box 4000, coin 5000

    talkData.Add(1000, new string[] { "안녕?:0", "반가워?:1", "조꾸르는 처음이지?:2" });
    talkData.Add(2000, new string[] { "누구나 난:3", "아 새인자구나:2", "이곳은 나의 땅이야:2", "구경할래?:1" });
    talkData.Add(3000, new string[] { "누군가가 사용했던 흔적이 있는 책상이다." });
    talkData.Add(4000, new string[] { "신비한 나무상자다", "무엇이 들어가 있을까? " });

    portraitData.Add(1000 + 0, portraitArr[0] );
    portraitData.Add(1000 + 1, portraitArr[1] );
    portraitData.Add(1000 + 2, portraitArr[2] );
    portraitData.Add(1000 + 3, portraitArr[3] );
    portraitData.Add(2000 + 0, portraitArr[4] );
    portraitData.Add(2000 + 1, portraitArr[5] );
    portraitData.Add(2000 + 2, portraitArr[6] );
    portraitData.Add(2000 + 3, portraitArr[7] );

    //quest talk
    talkData.Add(10 + 1000, new string[] { "어서와 :0", "이 마을에 놀라운 전설이 있어!:1", "왼쪽 호수 쪽에 루도가 알려줄거야.:2" });
    talkData.Add(11 + 1000, new string[] { "아직 못 만났어?:1", "루도는 왼쪽 호수에 있어.:1" });
    talkData.Add(11 + 2000, new string[] { "누구나 난:3", "아, 호수의 전설을 들으러 온거야?:1", "그럼 일 좀 하나 해주면 좋겠는데...:0", "내 집 근처에 떨어진 동전 좀 주워줬으면 해.:1" });

    talkData.Add(20 + 1000, new string[] { "루도의 동전?:1", "돈을 또 숨렸단말이야?:3", "루도 이자식 혼술이 나뉘야해.:3" });
    talkData.Add(20 + 2000, new string[] { "동전... 내 동전... 어딴지...:1" });
    talkData.Add(20 + 5000, new string[] { "루도의 동전을 발견했다." });
    talkData.Add(21 + 2000, new string[] { "찾아줘서 고마워!:2" });
}
```

참조 3개

```
public string GetTalk(int id, int talkIndex)
{
    if (!talkData.ContainsKey(id))
    {
        Debug.Log("없어" + talkIndex + "," + id);

        //퀘스트 맨 처음 대사마져 없을 때
        //기본 대사를 가지고 온다.
        if (!talkData.ContainsKey(id - id % 10))
            return GetTalk(id - id % 100, talkIndex);
        //해당 퀘스트 진행 중 대사가 없을 때
        //퀘스트 맨처음 대사를 가지고 온다.
        else
            return GetTalk(id - id % 10, talkIndex);
    }

    //끝나면 null 리턴
    if (talkIndex == talkData[id].Length)
    {
        return null;
    }
    else
        return talkData[id][talkIndex];
}
```

QuestData

```
참조 3개
public class QuestData
{
    public string questName;
    public int[] npcId;

    참조 3개
    public QuestData(string name, int[] npc)
    {
        questName = name;
        npcId = npc;
    }
}
```

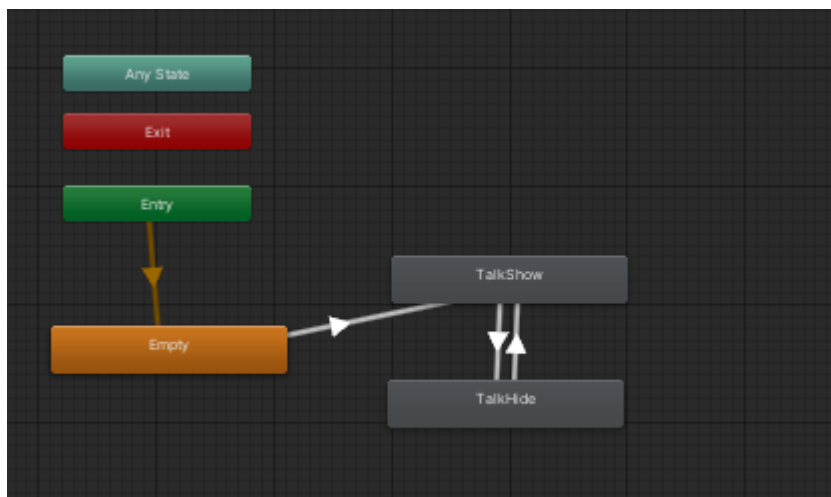
B25. 대화 애니메이션 느낌있게 만들기

1. 대화창 이펙트

대화창 관련 Animator controller, animation을 만든다.

isShow만든다. 애니메이션 2개 생성하여 드래그

대화UI를 화면아래로 안보이게 설정. 위치로 애니메이션을 만든다.

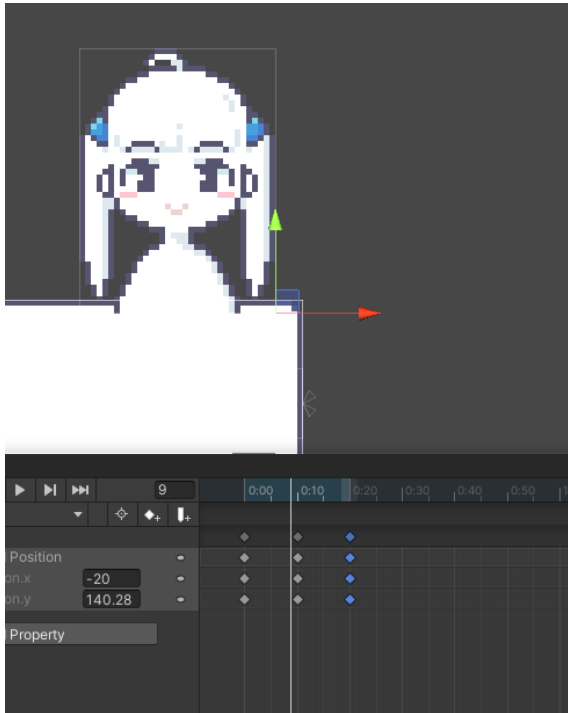


isShow는 gamemanager에서 설정한다. setActive 대신 SetBool을 해준다.

```
//visible talk for action
talkPanel.SetBool("isShow", isAction);
```

2. 초상화 이펙트

애니메이터, 애니메이션 1개씩 초상화 image에 생성. trigger사용



기존 Sprite와 다를 때 실행. 기존 sprite를 저장해서 비교후 애니메이션 실행

```
//portrait animation
if(prevSprite != portraitImg.sprite)
{
    portraitAnim.SetTrigger("doEffect");
    prevSprite = portraitImg.sprite;
}
```

3. 타이핑 이펙트

스크립트를 만들어

필요한 것. 메시지, 재생속도를 위한 변수 천

애니메이션 재생을 위한 시작, 재생, 종료 3개 함수 생성

Text 변수 생성, 초기화, 시작함수에서 공백 처리. Index는 글자 수를위해.

시간차 반복을 위해 invoke사용. Target과 text가 같으면 end.

시간차에서 확실한 소수값을 얻기위해 분자를 1.0f 작성

기존 text변수를 effectscrip로 변경

```

talk.SetMsg(talkData.Split(':')[0]);
//show portrait
portraitImg.sprite = talkManager.GetPortrait(id, int.Parse(talkData.Split(':')[1]));
portraitImg.color = new Color(1, 1, 1, 1);

//porrit animation
if(prevSprite != portraitImg.sprite)
{
    portraitAnim.SetTrigger("doEffect");
    prevSprite = portraitImg.sprite;
}

else

talk.SetMsg(talkData);

//hideportrait
portraitImg.color = new Color(1, 1, 1, 0);

```

사운드 : 오디오소스, 오디오 클립

AudioSource 변수를 생성, 초기화 후 재생함수에서 play()

```

//sound
if(targetMsg[index] != ' ' || targetMsg[index] != '.')
    audioSource.Play();

index++;

```

플레이가 되고 있을 때 현재 대사를 채워줘야한다.

애니메이션 실행 판단을 위한 플래그 변수 생성

플래그 변수를 이요해서 분기점 로직 작성

```

}
참조 2개
public void SetMsg(string msg)
{
    if (isAnim)
    {
        //interrupt
        msgText.text = targetMsg;
        CancelInvoke();
        EffectEnd();
    }
    else

```

매니저에서도 플래그 변수 사용해서 분기점 로직 작성

```
int questTalkIndex = 0;
string talkData = "";

//set talk data
if (talk.isAnim)
{
    talk.SetMsg("");
    return;
}

else
{
    questTalkIndex = questManager.GetQuestTalkIndex(id);
    talkData = talkManager.GetTalk(id + questTalkIndex, talkIndex);
}
```