

B20 : 찡꾸르 같은 탑다운 RPG 쉽게 준비하기

### 1. 플러그인 설치

Asset store에서 import!

2. Role Tile : 규칙을 정할 수 있는 타일 -> 따로 가져와야함

Rules는 내 주변에 있냐 없냐임!

3. 애니메이션 타일 -> rultile에서 output을 animation으로 프레임 개수대로 size

4. 외부 경계 : Tilemap으로

Collider 추가 -> tilemap, composite, rigid

Rigidbody body type : static

Tilemap collider : used by composite

안보이게 : mask interaction을 빌려.

Collider혹시 안보이면 암거나 한번 누르자

### 5. 픽셀 퍼펙트

Package manager에서 2D pixel perfect 다운

Pixel per unit과 해상도 맞춰야함

B21 : 찡꾸르식 액션 구현하기

### 1. 플레이어 십자이동

찡꾸르는 대각선 이동제한필요

Flag를 만들어서 수평, 수직 이동 제한.

### 2. 애니메이션

방향전환이 바로 되도록 any state.

하나의 방향에 서있기, 걷기 두개의 state로 구성

Transition연속적으로 태우면 애니메이션 작동안함. If문과 방향 변화 매개 변수 사용.

Can transition to self 해제

현재 asixRaw에 따라 수평 수직 판단하여 해결

### 3. 조사 액션

조가 가능한 오브젝트를 다른 Layer로 설정

#### B22. 대화창 UI 구축하기

Canvas pixel perfect

##### 1. 대화창 UI

사진 늘려도 괜찮으려면 image type을 sliced로. 그걸 위해서는 sprite 설정 필요

Sprite editor에서 모서리 잡기 위해 border값 입력

alt까지하면 이동. 꺾쳐우면 left, right여백이 생김

##### 2. 데이터 전달

UI프로그래밍을 위해서는 UnityEngine.UI using해야함

인스펙터 창에서 public 변수는 잊지말고 채워야한다.

##### 3. 상태 전환

액션 취할때만 대화창 나오게

SetActive()로 나올지말지 정할 수 있다.

상태 전환용 변수.

플레이어가 움직여서 이탈할 우려가있다. -> flag이용

set native size 원본 크기대로 이미지 사이즈 맞춤

애니메이션 : Animator 추가. Animation controller 추가. animator에서 new state 추가.

Window -> animation해서 애니메이션 적용된 오브젝트 선택하면 애니메이션 편집 가능

Anchor 적용된 UI는 anchored position 선택

#### B23 대화 시스템 구현하기

##### 1. 오브젝트 관리

스크립트 하나 만들어서 id, isNpc를 만들고 각각의 object들에 script를 넣어준다.

##### 2. 대화 시스템

대화 데이터를 관리할 매니저 생성

대화데이터를 저장할 Dictionary 변수 생성. 한쌍으로 이루어짐. Add를 통해서 추가가능.

지정된 대화 문장을 반환하는 함수 하나 생성. 아이디로 대화 get, talkindex로 한문장씩

이 talkManager을 gamemanager에서 선언하여 함수를 만들어 사용한다. Id는 objdata를 사용하여 scanobject를 받는다.

여기서 문제. Action이 켜져있으면 꺼져있도록 해놨는데 그러면 대사가 하나만 나온다!

- ➔ 대화가 모두 끝나야 액션이 끝나도록 한다. talkIndex와 문장갯수를 비교하여 끝을 확인한다. 다음문장을 위해서는 talkindex++, 이야기가 끝나면 0으로 초기화

## 2. 초상화

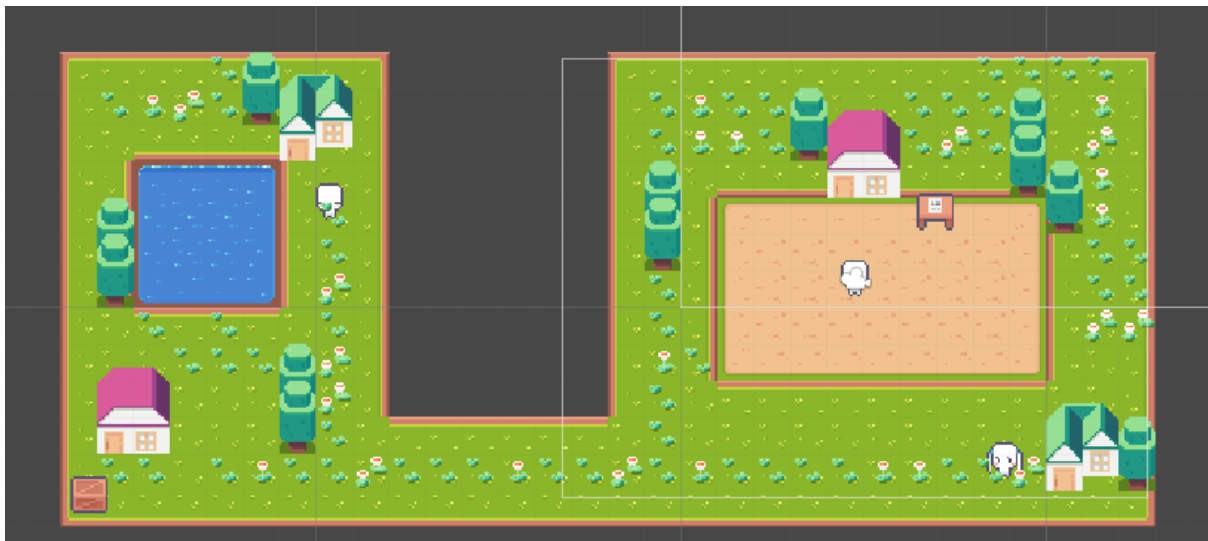
이미지를 넣고. GameManger에 변수 생성, 할당.

대화별로 sprite다르게 해준다. -> dictionary사용. Spite 배열 사용 id + 1이런식으로 하면된다.

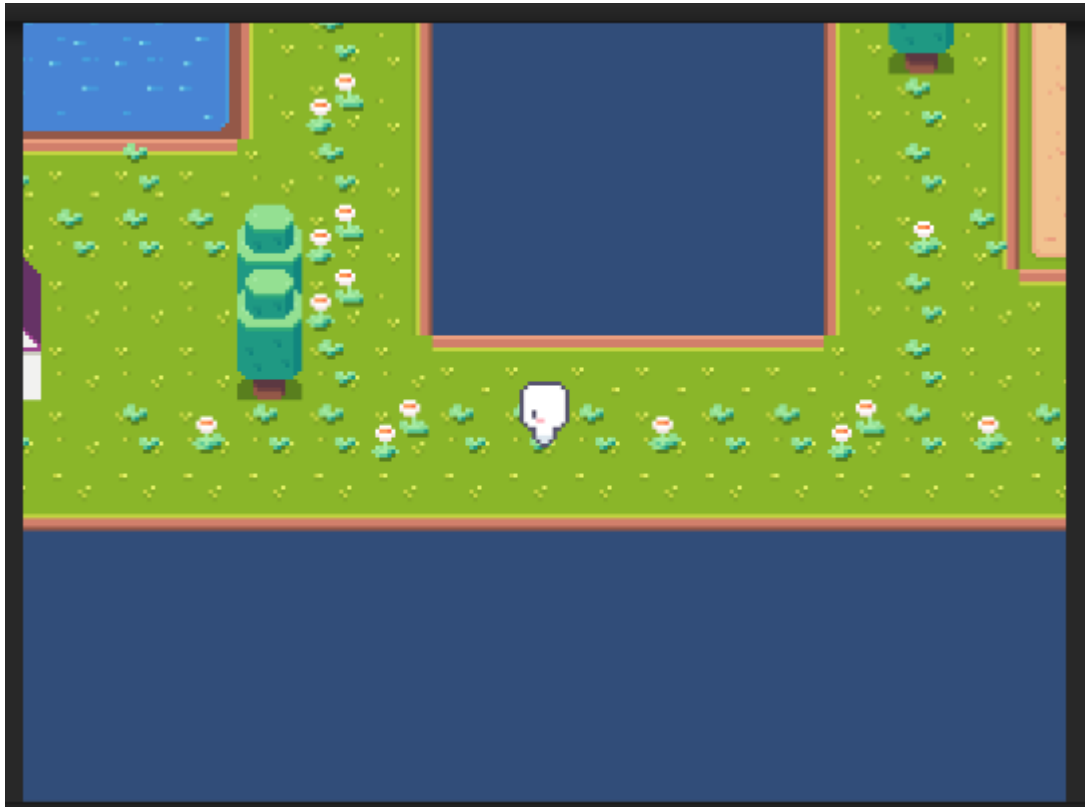
초상화 가져올 함수 만들어준다. 표정은 문장과 1:1 매칭 방법은 :등 구분자를 통해 문장 뒤에 숫자를 적어준다. Split을통해 자르고. string으로 나누기 때문에 int.parse()를 통해 형변환을 해준다.

사진들

전체 맵



옆모습



object



npc



## TalkManager

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

// Unity 스크립트 | 참조 1개
public class TalkManager : MonoBehaviour
{
    Dictionary<int, string[]> talkData;
    Dictionary<int, Sprite> portraitData;

    public Sprite[] portraitArr;

    // Unity 메시지 | 참조 0개
    private void Awake()
    {
        talkData = new Dictionary<int, string[]>();
        portraitData = new Dictionary<int, Sprite>();
        GenerateData();
    }

    // 참조 1개
    void GenerateData()
    {
        talkData.Add(1000, new string[] { "안녕?:0", "반가워?:1", "쯔꾸르는 처음이지?:2" });
        talkData.Add(2000, new string[] { "누구나 넌:3", "아 새입자구나:2", "이곳은 나의 땅이야:2", "구경할래?:1" });
        talkData.Add(100, new string[] { "누군가가 사용했던 흔적이 있는 책상이다." });
        talkData.Add(200, 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]);
    }

    // 참조 1개
    public string GetTalk(int id, int talkIndex)
    {
        if(talkIndex == talkData[id].Length)
        {
            return null;
        }
        else
        {
            return talkData[id][talkIndex];
        }
    }

    // 참조 1개
    public Sprite GetPortrait(int id, int portraitIndex)
    {
        return portraitData[id + portraitIndex];
    }
}
```

## objectData

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

// Unity 스크립트 | 참조 2개
public class ObjectData : MonoBehaviour
{
    public int id;
    public bool isNpc;
}
```

## GameManager

```
public class GameManager : MonoBehaviour
{
    public TalkManager talkManager;
    public GameObject talkPanel;
    public Text talkText;
    public GameObject scanObject;
    public Image portraitImg;
    public bool isAction;
    public int talkIndex;
    // Start is called before the first frame update

    @ Unity 메시지 | 참조 0개
    private void Awake()
    {
        talkPanel.SetActive(false);
    }

    참조 1개
    public void Action(GameObject scanObj)
    {
        scanObject = scanObj;
        ObjectData objectData = scanObj.GetComponent<ObjectData>();
        Talk(objectData.id, objectData.isNPC);

        talkPanel.SetActive(isAction);
    }

    참조 1개
    void Talk(int id, bool isNPC)
    {
        string talkData = talkManager.GetTalk(id, talkIndex);

        if (talkData == null)
        {
            isAction = false;
            talkIndex = 0;
            return;
        }

        if (isNPC)
        {
            talkText.text = talkData.Split(':')[0];
            portraitImg.sprite = talkManager.GetPortrait(id, int.Parse(talkData.Split(':')[1]));
            portraitImg.color = new Color(1, 1, 1, 1);
        }
        else
        {
            talkText.text = talkData;
            portraitImg.color = new Color(1, 1, 1, 0);
        }

        isAction = true;
        talkIndex++;
    }
}
```

## PlayerAction

```

@ Unity 메시지 | 참조 0개
void Awake()
{
    rigid = GetComponent<Rigidbody2D>();
    anim = GetComponent<Animator>();
}

// Update is called once per frame
@ Unity 메시지 | 참조 0개
void Update()
{
    // move value
    h = manager.isAction ? 0 : Input.GetAxisRaw("Horizontal");
    v = manager.isAction ? 0 : Input.GetAxisRaw("Vertical");

    //check button down & up
    bool hDown = manager.isAction ? false : Input.GetButton("Horizontal");
    bool vDown = manager.isAction ? false : Input.GetButton("Vertical");
    bool hUp = manager.isAction ? false : Input.GetButtonUp("Horizontal");
    bool vUp = manager.isAction ? false : Input.GetButtonUp("Vertical");

    //check horizontal move
    if (hDown)
        isHorizonMove = true;
    else if (vDown)
        isHorizonMove = false;
    else if (hUp || vUp)
        isHorizonMove = h != 0;

    //animation
    if(anim.GetInteger("hAxisRaw") != h)
    {
        anim.SetInteger("hAxisRaw", (int)h);
        anim.SetTrigger("isChange");
    }

    else if(anim.GetInteger("vAxisRaw") != v)
    {
        anim.SetInteger("vAxisRaw", (int)v);
        anim.SetTrigger("isChange");
    }

    //Direction
    if (vDown && v == 1)
        direcVec = Vector3.up;
    else if (vDown && v == -1)
        direcVec = Vector3.down;
    else if (hDown && h == -1)
        direcVec = Vector3.left;
    else if (hDown && h == 1)
        direcVec = Vector3.right;

    //scan object
    if(Input.GetButtonDown("Jump") && scanObject != null)
    {
        manager.Action(scanObject);
    }
}

```



@ Unity 메시지 | 참조 0개

```
private void FixedUpdate()
```

```
{
```

```
    //move
```

```
    Vector2 moveVec = isHorizonMove ? new Vector2(h, 0) : new Vector2(0, v);
```

```
    rigid.velocity = moveVec * Speed;
```

```
    //Ray
```

```
    Debug.DrawRay(rigid.position, direcVec * 0.7f, new Color(0, 1, 0));
```

```
    RaycastHit2D rayHit = Physics2D.Raycast(rigid.position, direcVec, 0.7f, LayerMask.GetMask("Object"));
```

```
    if (rayHit.collider != null)
```

```
        scanObject = rayHit.collider.gameObject;
```

```
    else
```

```
        scanObject = null;
```

```
}
```