

TODO 봄학점인 구현에 있어서는 BGM 쓸만한가 2024.05.11 목

1. Input - Input을 헤론에서 관리
2. Scene - 신호링에서 표기하기
3. Data - JSON 파일 읽기
4. Sound - BGM, 효과음(oneshot)
5. UI - 표기, 고정 나누어서 고정

1. Input

- 가능 기능이나 보통 입력을 한곳에서 처리.

心底(거의)로 이를, InputManager의 Update()에서
이루고 싶다.

```
public class InputManager
{
    public Action KeyAction = null;
    public void UpdateInput()
    {
        if (Input.anyKey == false) return;
        if (KeyAction != null)
            KeyAction.Invoke();
    }
}
```

2. Scene

I - 신호기로

상호작용하여 사용

제한된 몇몇 개의

```
public abstract class BaseScene : MonoBehaviour
{
    public Define.Scene _sceneType = Define.Scene.Unknown;
    protected void Awake()
    {
        Init();
    }
    protected virtual void Init()
    {
        // 초기화 할 것
    }
    public abstract void Clear();
}
```

```
public enum Scene
{
    Unknown,
    Login,
    Lobby,
    Game
}
```

II - 신호기로.

수반되는 enum은 구조화된 형태.

```
public class SceneManagerEx
{
    // 현재 씬
    public BaseScene CurrentScene { get { return GameObject.FindObjectOfType<BaseScene>(); } }

    // Define.Scene에 있는 씬 로드하기
    public void LoadScene(Define.Scene type)
    {
        Managers.Clear(); 클리어!
        SceneManager.LoadScene(GetSceneName(type));
    }

    // Define.Scene으로 이름찾기
    string GetSceneName(Define.Scene type)
    {
        string name = System.Enum.GetName(typeof(Define.Scene), type);
        return name;
    }

    // 초기화
    public void Clear()
    {
        CurrentScene.Clear();
    }
}
```

3. Data

Json 읽기와 딕셔너리 만들기.

Data Class 만들고

ILoader 상속받아

Make Dict 실행.

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

// 요거 상속받아서 만들기
public interface ILoader <Key, Value>
{
    Dictionary<Key, Value> MakeDict();
}

public class DataManager
{
    //EX)
    //public Dictionary<int, Data.Skill> SkillDict { get; private set; } = new Dictionary<int, Data.Skill>();

    public void Init()
    {
        //SkillDict = LoadJson<Data.SkillData, int, Data.Skill>("StatData").MakeDict();
    }

    Loader LoadJson<Loader, Key, Value>(string path) where Loader : ILoader <Key, Value>
    {
        TextAsset textAsset = Managers.Resource.Load<TextAsset>($"Data/{path}");
        return JsonUtility.FromJson<Loader>(textAsset.text);
    }
}
```

4. Sound (Audio)

BGM, Onshot

3D 소리

AudioSource - BGM, Onshot 음악

AudioClip - <name, Clip> 딕셔너리로 관리

AudioListener - 헤어리터

Init - BGM을 초기화드 만들고 실행

Play (Clip, type, pitch)

Clip은 type에 따라 다르며, pitch로 설정

Get or Add AudioClip - 초기화를 들고나와서 추가하거나 개설하기

Clear - 헤어리터 초기화.

ToDo - 3D Sound..

Play Clip At point

5. UI - 자동화

UI ~ 자식찾아서 바인딩해 주고
자식의 텍스트라든가 속성 찾으기
자식 블록이기, 소멸하기

이벤트 청들어 만들어서 드래그/클릭 구현

호출창 제작 UI 모듈
PopUp, Scene UI 3분
제작자. 제작자 상수유지