

Platformer Jumping Game 포트폴리오

개요

- 플랫폼머 점프 게임으로 맵의 다양한 장애물을 돌파하여 최종점에 오르면 되는게임입니다
- 플랫폼 : windows
- 참여 인원 : 2명
- 역할 : 메인 프로그래머 및 기획
- 개발 기간 : 2021.9. ~ 2022.3.(9개월)

개발 내용

- 주요개발 내용
 - 플레이어 움직임, 상호작용, 장애물 제작
 - UGUI를 이용한 UI 설계 및 구현
 - JsonUtility를 통한 저장/로드 시스템
 - 풋 스텝 사운드 재생 설계
 - 카메라 이동, 지형물 투영화 처리
 - Timeline 시스템을 활용한 엔딩 시네마틱 영상 제작
- 기타
 - 인게임 시네마틱 영상 제작
 - 플레이어 애니메이션 로직 구현

성과

- 기본 Unity 프로그래밍 숙달
- Facefunch.steamworks 라이브러리 사용 경험
- UnityGUI, JsonUtility, Cinemachine의 사용 경험
- 프레임 최적화를 위한 비동기 프로그래밍 구현 경험

플레이 스크린 샷

메인 화면



사다리 타고있는 모습



장애물에 맞아 날아가는 모습



상들리에를 통과하는 모습



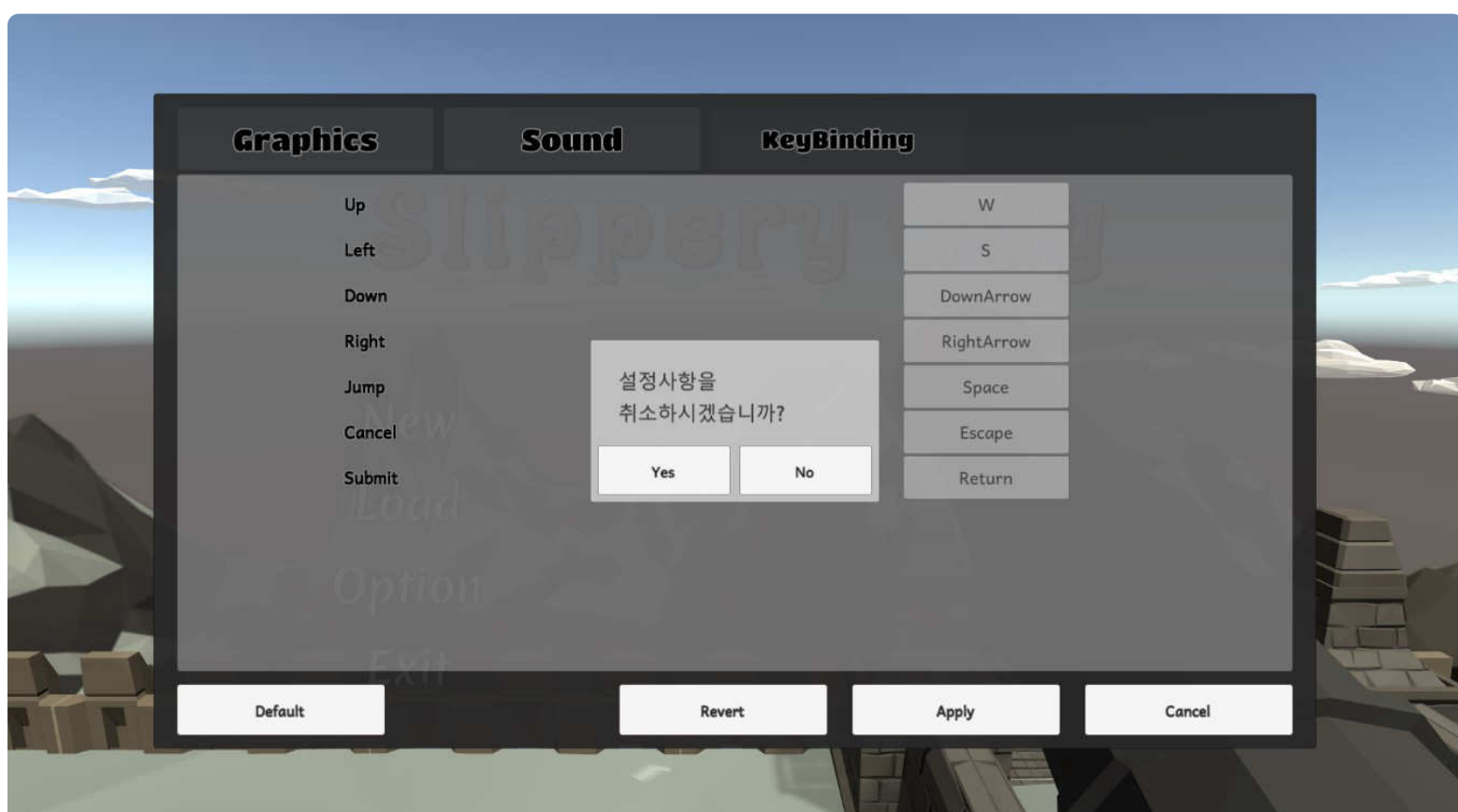
화살에 맞아 낙백 당하는 모습



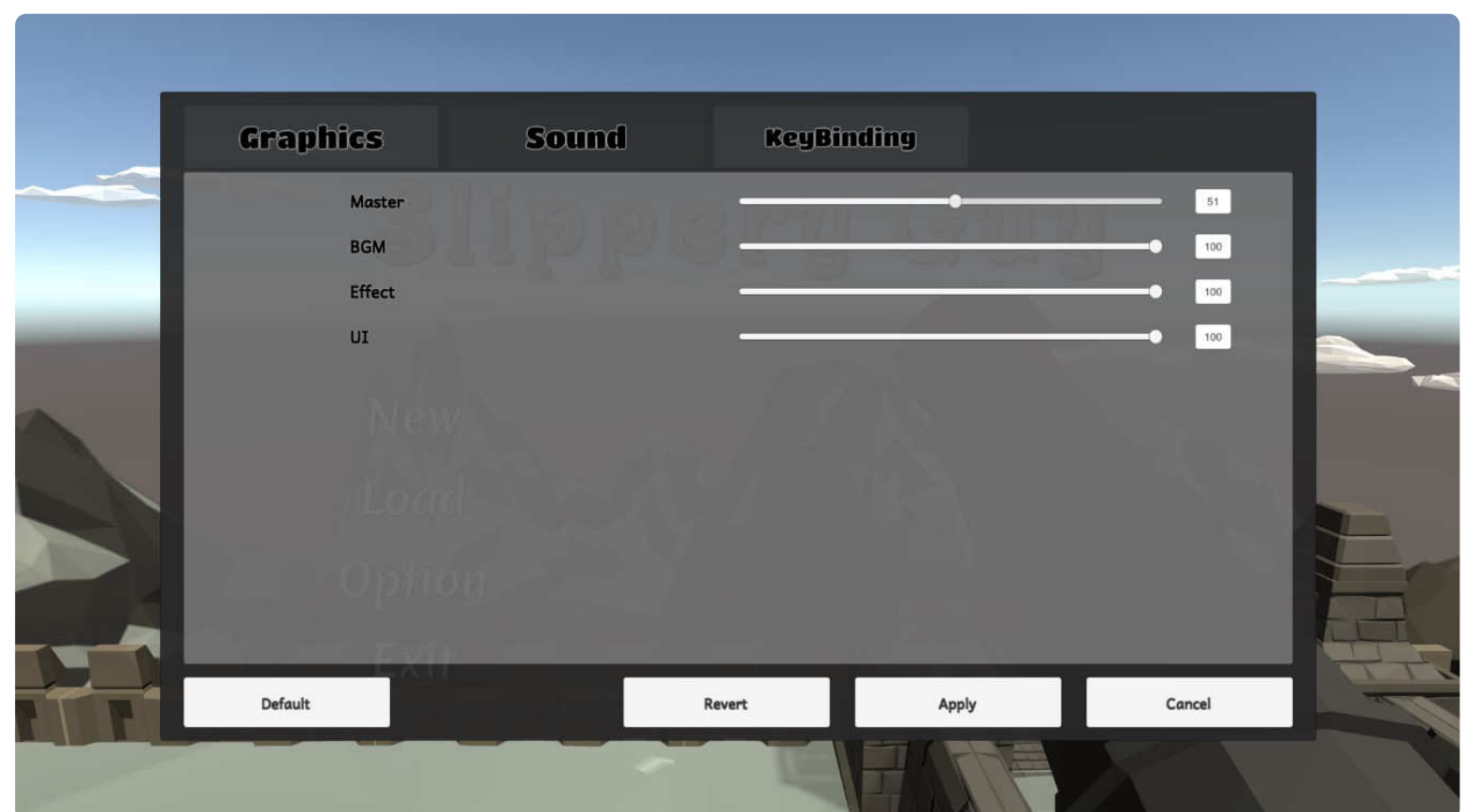
바람을 타고 위로 올라가는 모습



키 바인딩 과 응답창



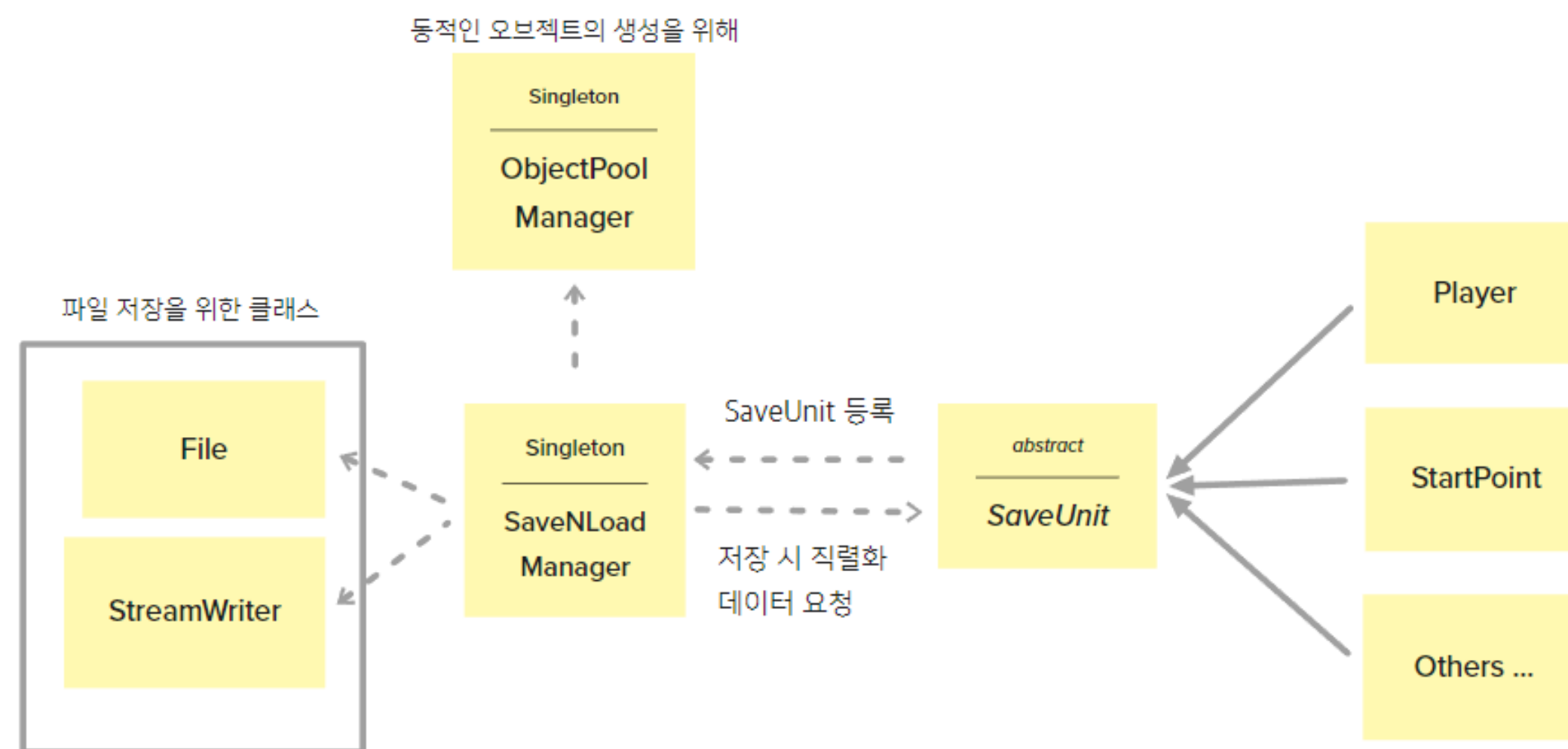
사운드 조절창



저장시스템 개요 및 코드

개요

- 저장시스템은 SaveNLoadManager와 SaveUnit의 발행 구독 패턴의 아이디어로 설계된 시스템입니다.
- 데이터의 저장은 어떤 string 데이터를 사용하여도 무방합니다. 다만 각 객체의 데이터를 저장하기 위한 추가 로직이 필요하게 되므로, JsonUtility를 통해 편리하게 객체<-> Json로 전환하였습니다.
- 저장이 필요 객체는 SaveUnit을 상속받은 이후, GetSavingData, SetSaveData를 Overwriting 하여 구현
- 객체 데이터를 string으로, string 데이터를 객체 데이터로 각각 바꾸어 처리하면 됩니다.



저장시스템 모식도

프로그램 코드 일부

```
// 게임을 저장할때 사용 ...  
참조 1개  
public async void Save()  
{  
    using (StreamWriter sw = File.CreateText(tempPath))  
    {  
        //정적인 객체들 저장  
        for (int i = 0; i < staticData.Count; i++)  
            AddSaveDataToStringBuilder(staticData[i]);  
  
        //정적데이터 동적데이터를 나누기 위한 splitter  
        sb.Append(staticLineSplitter).Append(dataSplitter);  
  
        //동적인 객체들 저장  
        for (int i = 0; i < dynamicData.Count; i++)  
            AddSaveDataToStringBuilder(dynamicData[i]);  
  
        byte[] allData = Encoding.Default.GetBytes(sb.ToString());  
  
        sb.Clear();  
        //sw.Write(Convert.ToBase64String(allData));  
        await sw.WriteAsync(Convert.ToBase64String(allData));  
    }  
  
    if (File.Exists(dataPath))  
        File.Delete(dataPath);  
    File.Move(tempPath, dataPath);  
}  
  
#if UNITY_EDITOR  
    Debug.Log("SaveComplete");  
#endif  
}  
  
// 오브젝트를 식별하기 위한 데이터를 추가  
참조 2개  
private void AddSaveDataToStringBuilder(SaveUnit unit)  
{  
    string uniqueName;  
    object data = unit.GetSavingData(out uniqueName);  
    sb.Append(uniqueName).Append(nameSplitter).Append(JsonUtility.ToJson(data)).Append(dataSplitter);  
}
```

Save 코드

```
// 데이터를 로드할 때 사용  
참조 1개  
public void Load()  
{  
    try  
    {  
        string encodedData;  
        string convertData;  
        using (StreamReader sr = File.OpenText(dataPath))  
        {  
            encodedData = sr.ReadToEnd();  
  
            convertData = Encoding.Default.GetString(Convert.FromBase64String(encodedData));  
            string[] splittedData = convertData.Split(dataSplitter);  
  
            GameObject[] staticDataObject = GameObject.FindGameObjectsWithTag("StaticSave");  
            int i;  
            //정적 객체로드  
            for (i = 0; i < splittedData.Length; i++)  
            {  
                if (splittedData[i] == staticLineSplitter)  
                    break;  
                SetStaticData(splittedData[i], staticDataObject);  
            }  
            i += 1;  
            //동적 객체로드  
            for (; i < splittedData.Length - 1; i++)  
                SetDynamicData(splittedData[i]);  
        }  
    }  
    catch  
    {  
        DeActivate();  
    }  
}  
  
// ObjectPoolManager에 등록되어 사용되는 오브젝트에 데이터를 설정 때 사용  
// 즉 동적으로 생성이 필요한 오브젝트들에 사용  
참조 1개  
private void SetDynamicData(string data)  
{  
    string[] sep = data.Split(nameSplitter);  
    GameObject go = ObjectPoolManager.Instance.GetGameObject(sep[0]);  
    go.GetComponent<SaveUnit>().SetSaveData(sep[1]);  
}  
  
// StaticSave 태그가 붙은 오브젝트의 데이터를 설정할 때 사용  
참조 1개  
private void SetStaticData(string jsonData, GameObject[] staticData)  
{  
    string[] sep = jsonData.Split(nameSplitter);  
    GameObject go = Array.Find(staticData, x => x.name.CompareTo(sep[0]) == 0);  
    go.GetComponent<SaveUnit>().SetSaveData(sep[1]);  
}
```

Load 코드

더욱 자세한 사항은 아래 링크를 참고해주시면 됩니다.

- <https://github.com/shehdrbs123/shehdrbs123/tree/main/3DJumpGame/SaveSystemDetailFortfolio>