쿠키런 오븐브레이크

UNITY 부트캠프 17기 1조 신준 | 윤대석 | 조하은 | 현동호

01. 팀원 소개

신준(조장)

- Map 제작 총괄
- Spawn Manager & Items
- 팬케이크맛 쿠키 제작

윤대석

- 전반적 PlayerState
- Map & Enemy
- InGame UI

조하은

- UI/UX
- Sound Manager
- 달빛술사 쿠키 보조

현동호

- UI
- PlayerState
- BonusTime 담당
- 달빛술사 쿠키 제작

02. 프로젝트 소개

오븐에 구어질 위기에 빠진
다양한 맛들의 쿠키들..
*쿠키들의 **오븐탈출 대작전**이 시작된다!*

- 모바일 환경 최적화 게임
- 제작사: 데브시스터즈
- 장르: 사이드 스크롤 러닝 액션게임

- 목표

점프와 슬라이드로 조작을 하며 스크린의 장애물들을 피해 달리면서 젤리를 먹어 점수를 올리는 게임

03-1. 기술 설명

디자인 패턴_싱글톤 / StatePattern

Singleton(싱글톤)

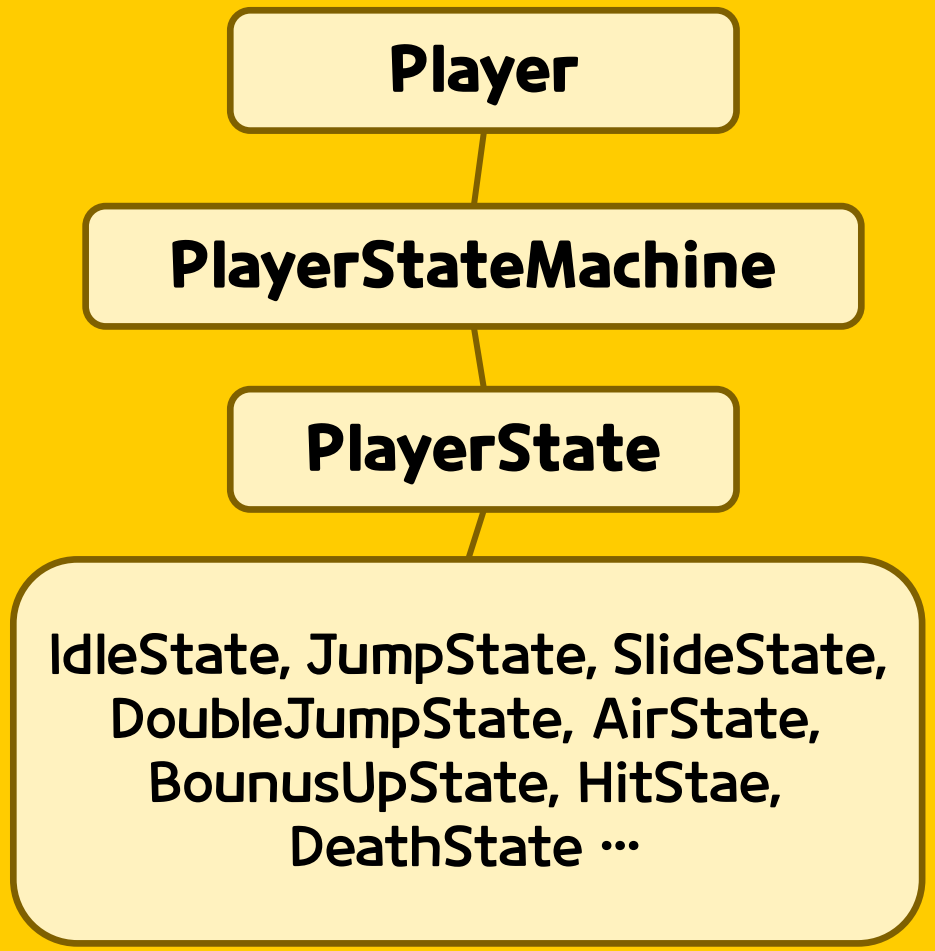
```
Unity 스크립트|참조 5개
public class SingletonBehaviour<T> : MonoBehaviour where T : MonoBehaviour
{
    private static T _instance;

    // code by 동호
    참조 99+개
    public static T Instance // 프로퍼티
    {
        get
        {
```

```
9   using UnityEngine;
10
        Unity 스크립트(자산 참조 2개)|참조 0개
11  public class PlayerManager : SingletonBehaviour<PlayerManager>
12  {
13      [SerializeField] Dictionary<int, GameObject> PlayerGameObjectDict = ne
14      [SerializeField] GameObject cookie_100;
15      [SerializeField] GameObject cookie_101;
```

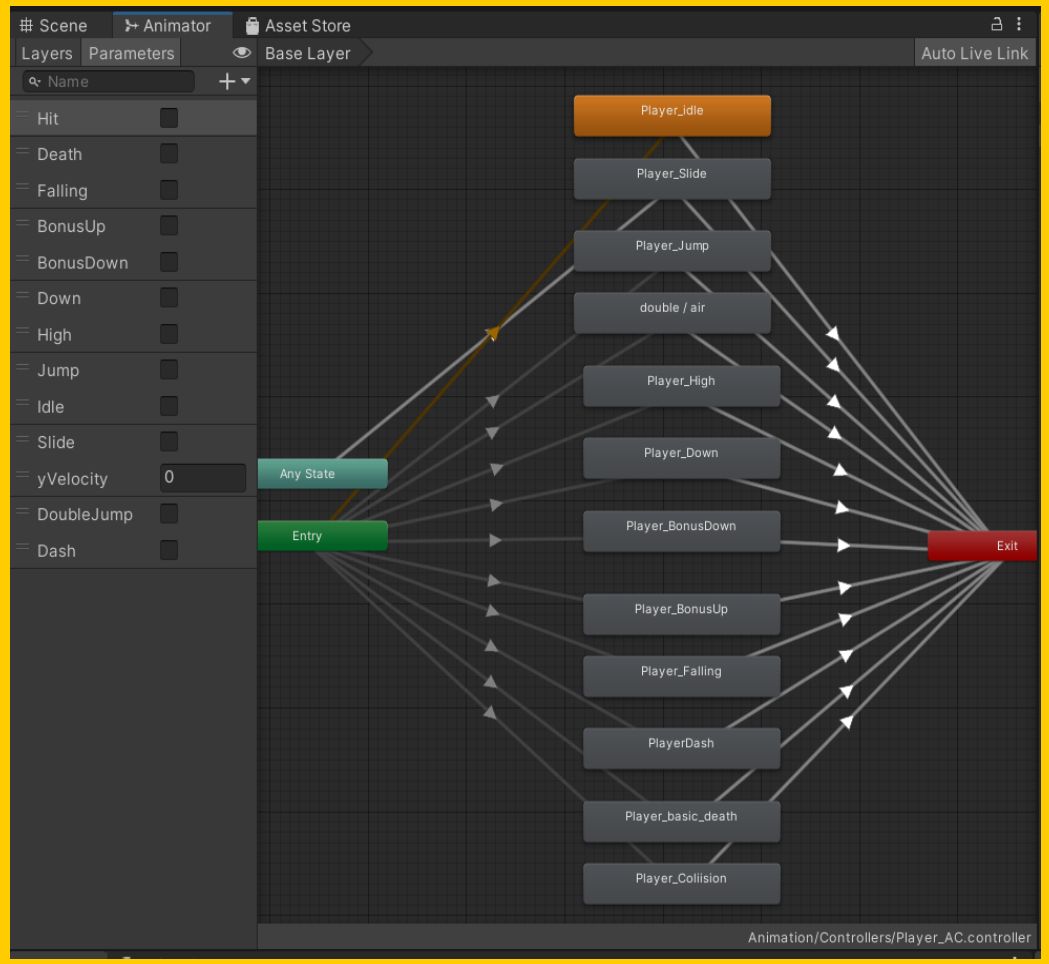
StatePattern_Player

상태패턴

: 객체 내부 상태에 맞춰
스스로 행동을 변경하는 패턴



StatePattern_Player



! 모바일 터치패드 활용 시 필요한 추가 코드 생성

```
참조 1개
void JumpButton(PointerEventData data)
{
    if (player.stateMachine.currentState == player.fallingState
        || player.stateMachine.currentState == player.highState
        || player.stateMachine.currentState == player.downState
        || player.stateMachine.currentState == player.deathState)
        return;

    if(player.isBonusStart == true)
    {
        if (player.gValue <= 0)
            return;
        BonusTimeButtonDown();
    }
    else
    {
        _buttonPush = true;

        if (player == null) return;

        if (player.IsGroundDetected())
        {
            isjumping = true;
            player.stateMachine.ChangeState(player.jumpState);
        }
    }

}
```

03-2. 기술 설명

기타 상세 기술

UI 자동화 환경 생성

```
Unity 스크립트 | 참조 3개
public class UI_Base : MonoBehaviour
{
    //code by.동호
    참조 1개
    public static void AddUIEvent(GameObject go, Action<PointerEventData> action, Define.UIEvent type = Define.UIEvent.Click)
    {
        UI_EventHandler evt = Util.GetOrAddComponent<UI_EventHandler>(go); // 자동으로 UI_EventHandler 부착 및 가져옴

        switch (type)
        {
            case Define.UIEvent.Click:
                evt.OnClickHandler -= action;
                evt.OnClickHandler += action; //구독 신청하고
                break;
            case Define.UIEvent.PointerDown:
                evt.OnPointerDownHandler -= action;
                evt.OnPointerDownHandler += action; //구독 신청하고
                break;
            case Define.UIEvent.PointerUp:
                evt.OnPointerUpHandler -= action;
                evt.OnPointerUpHandler += action; //구독 신청하고
                break;
        }

    }
}
```

UI 자동화 환경 생성

Ex. Player 조작 버튼 등에 사용

```
Unity 메시지 | 참조 0개
private void Start()
{
    //BonusJump.gameObject.AddUIEvent(ButtonClicked, type : Define.UIEvent.PointerDown);
    Jump.gameObject.AddUIEvent(JumpButton, type: Define.UIEvent.PointerDown); // 버튼 다운 했을 때 이벤트 등록 (타입의 기본값 : Define.UIEvent.Click)
    Jump.gameObject.AddUIEvent(JumpButtonUp, type: Define.UIEvent.PointerUp); // 버튼 다운 했을 때 이벤트 등록 (타입의 기본값 : Define.UIEvent.Click)
    Jump.gameObject.AddUIEvent(DoubleJump, type: Define.UIEvent.PointerDown);
    Jump.gameObject.AddUIEvent(DoubleJumpUP, type: Define.UIEvent.PointerUp);
    Pause.gameObject.AddUIEvent(PauseGame);
    KeepGame.gameObject.AddUIEvent(PauseCancle);
    ReStart.gameObject.AddUIEvent(ReStartDown);
    GiveUp.gameObject.AddUIEvent(GiveUP);

    Slide.gameObject.AddUIEvent(SlideButton, type: Define.UIEvent.PointerDown);
    Slide.gameObject.AddUIEvent(SlideButtonUp, type: Define.UIEvent.PointerUp);

    BonusJumpBtn.gameObject.AddUIEvent(OnButtonDown, type: Define.UIEvent.PointerDown); // 버튼 다운 했을 때 이벤트 등록 (타입의 기본값 : Define.UIEvent.Click)
    BonusJumpBtn.gameObject.AddUIEvent(OnButtonUp, type: Define.UIEvent.PointerUp); // 버튼 업 했을 때 이벤트 등록 (타입의 기본값 : Define.UIEvent.Click)

    BonusSlideBtn.gameObject.AddUIEvent(OnButtonDown, type: Define.UIEvent.PointerDown); // 버튼 다운 했을 때 이벤트 등록 (타입의 기본값 : Define.UIEvent.Click)
    BonusSlideBtn.gameObject.AddUIEvent(OnButtonUp, type: Define.UIEvent.PointerUp); // 버튼 업 했을 때 이벤트 등록 (타입의 기본값 : Define.UIEvent.Click)

    BonusJumpBtn.gameObject.SetActive(false);
    BonusSlideBtn.gameObject.SetActive(false);
```

Data Manager

활용:
- UI 동적 구현
- 캐릭터별 Sound 적용

Excel 시트 작성
(변수별 값 작성)

↓

.Json 파일로 변환

↓

Data Manager로
Data 연동하여 UI 동적 구현

id	name	sprite_name	price
int	string	string	int
100	용감한 쿠키	cookie01	10
101	팬케이크맛 쿠키	cookie02	20
102	달빛술사 쿠키	cookie03	30
103	앨리스 좀비맛 쿠키	cookie04	9999
104	뱀파이어맛 쿠키	cookie05	9999
105	탐험가맛 쿠키	cookie06	9999
106	천사맛 쿠키	cookie07	9999
107	생일케이크맛 쿠키	cookie08	9999

Data Manager

LOBBY UI_ScrollView

횡스크롤뷰로 옆으로 넘기면서 다양한 쿠키 선택 및 구매 가능

Data Manager

선택한 캐릭터가 LOBBY, Ingame, 결과창에 출력

```
참조 1개
private HE_DataManager() { }

참조 2개
public MycookiesData GetData(int id)
{
    return dicMycookiesData[id];
}
참조 8개
public void LoadData()
{
    //TextAsset을 로드(Resources폴더-Data폴더-Mycookies_data파일을 Load)
    TextAsset asset = Resources.Load<TextAsset>("Data/MyCookies_data");
    //asset의 문자열 출력
    var json = asset.text;

    //역직렬화
    MycookiesData[] arrMycookiesDatas = JsonConvert.DeserializeObject<MycookiesData[]>(json);

    //key를 id로 해서 새로운 사전 객체가 생성되어 반환됨(for문에서 dicMycookiesData(data.id, data)해준 것과 같은 기능)
    dicMycookiesData = arrMycookiesDatas.ToDictionary(x => x.id);
}

//dicMycookiesData의 Values를 List로 반환
참조 8개
public List<MycookiesData> GetMycookiesDatas()
{
    return dicMycookiesData.Values.ToList();
}
```

Data Manager

Ex) LOBBY_Main스크립트
DataManager에서 해당 id의 쿠키 data를 받아서 data.sprite_name을 Lobby에 출력

```
void Start()
{
    //데이터 로드
    HE_DataManager.instance.LoadData();
    MycookiesData data = HE_DataManager.instance.GetMycookiesDatas().Find(cookie => cookie.id == UserDataManager.Instance.GetSelectCookieID());
    string cookieName = string.Empty;
    if(data != null)
    {
        var atlas = AtlasManager.Instance.GetAtlasByName("Cookies");
        cookieName = data.sprite_name;
        if (!string.IsNullOrEmpty(cookieName))
        {
            //MYCOOKIES에서 선택한 쿠키의 spirte를 LOBBY에 출력
            LobbyCookieImg.sprite = atlas.GetSprite(cookieName);
            LobbyCookieImg.gameObject.transform.localScale = new Vector2(2f, 2f);
        }
    }
```

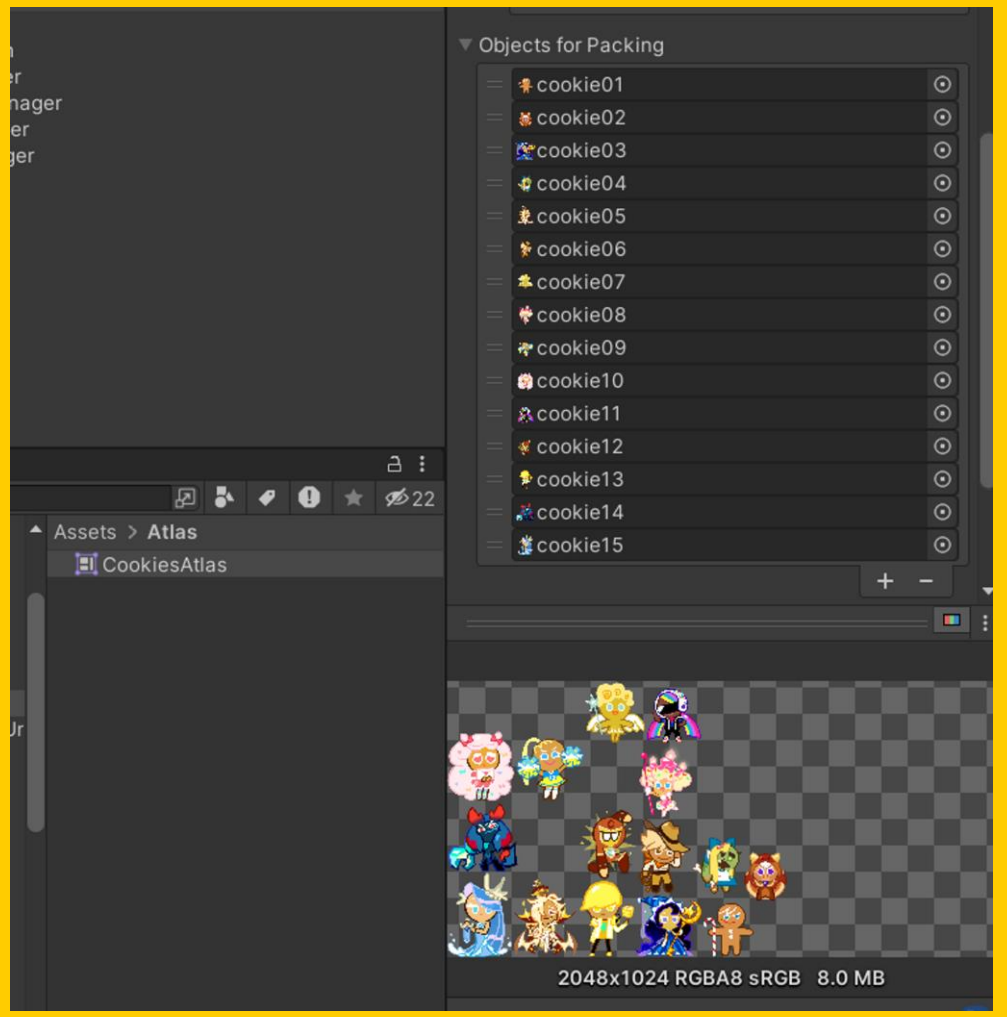
Data Manager

선택한 캐릭터가 LOBBY, Ingame, 결과창에 출력

Atlas Manager

Atlas

: 여러 개의 텍스처를 단일 텍스처로 결합하는 에셋
: Unity가 여러 개의 드로우 콜을 발행하는 대신
이러한 단일 텍스처를 호출함으로써 하나의 드로우 콜을 발행



UserDataManager

현재 Player가 보유한 쿠키의 데이터 관리

: 보유 코인과 쿠키 가격 비교 후 팝업창 출력
: 캐릭터 해금 및 선택에 UserDataManager 사용

Spawn Manager

구글스프레드 시트 작성
(변수별 값 작성)

↓

.Json 파일로 변환

↓

Spawn Manager로
Data 연동

index	JellyType	JellyYpos	JellyAmount	ObstacleType	Obstacle	Ground
40			1			
41			1			
42			1	1	1	3
43			1	1	1	
44			1	1	1	3
45			1	1	1	
46			1	1	1	3
47			1	1	1	
48			1	1	1	3
49			1	1	1	
50			1	1	1	
51	8		1			
52	9		1			
53	10		1			
54	11		1			
55	12		1			

Spawn Manager

Ex. 젤리타입


```
string jsonFilePath = CurrentMap; //리소스 파일 안에 있는 data읽기

if (File.Exists(jsonFilePath)) //파일이 존재할 시
{
    // JSON 파일 읽기
    string jsonContent = File.ReadAllText(jsonFilePath);
    JsonData jsonData = JsonUtility.FromJson<JsonData>(jsonContent); //Json으로 읽은 데이터 Class에 대입


    // 출력

    if (Time.time > LastSpawnTime + SpawnSpeed / p.GroundScrollSpeed && patternNum < jsonData.index.Count)//몬스터 생성 주기
    {
        LastSpawnTime = Time.time;

        StartCoroutine(spawn());
        jellytype = jsonData.JellyType[patternNum];      //데이터 가져오기 코루틴으로 쓰기 위해 대입해줌
        jellyYpos = jsonData.JellyYpos[patternNum];      //데이터 가져오기 코루틴으로 쓰기 위해 대입해줌
        obstacleType = jsonData.ObstacleType[patternNum];    //데이터 가져오기 코루틴으로 쓰기 위해 대입해줌
        jellyAmount = jsonData.JellyAmount[patternNum];      //데이터 가져오기 코루틴으로 쓰기 위해 대입해줌
        obstacle = jsonData.Obstacle[patternNum];            //데이터 가져오기 코루틴으로 쓰기 위해 대입해줌
        ground = jsonData.Ground[patternNum];
        patternNum++; //index값과 같음.

    }

}
```

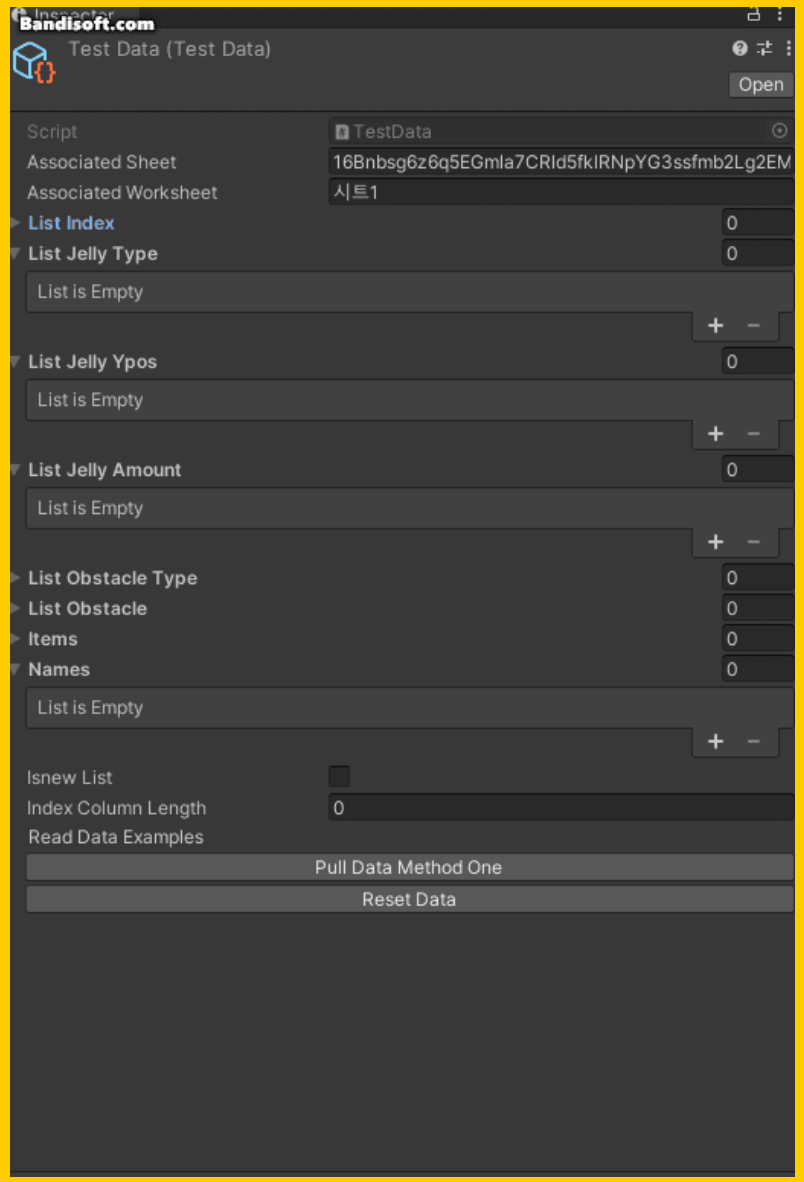
각 변수에 해당 값(Prefab 등)을 동적으로 생성
생성 주기와 Map Scroll Speed 동기화

Spawn Manager

맵을 수정할 때 마다
Json파일을 수정해야하는 번거로움 발생

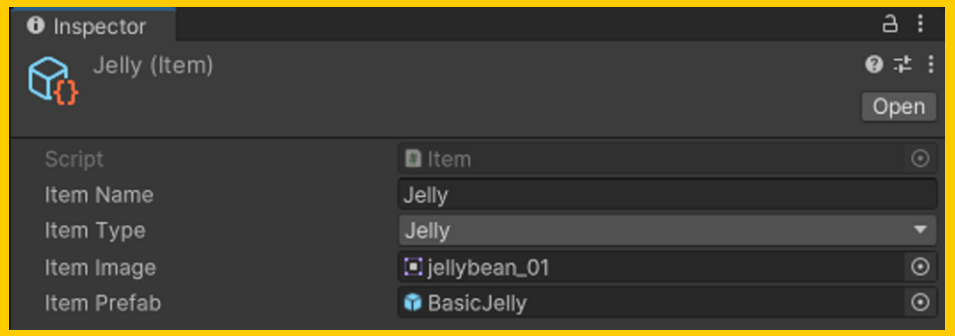
GSTU 플러그인을 활용하여
Unity와 스프레드 시트를 동기화하여 자동수정 환경 구축

But,
고정된 Index Count값만 연동된다는 문제점 발생..

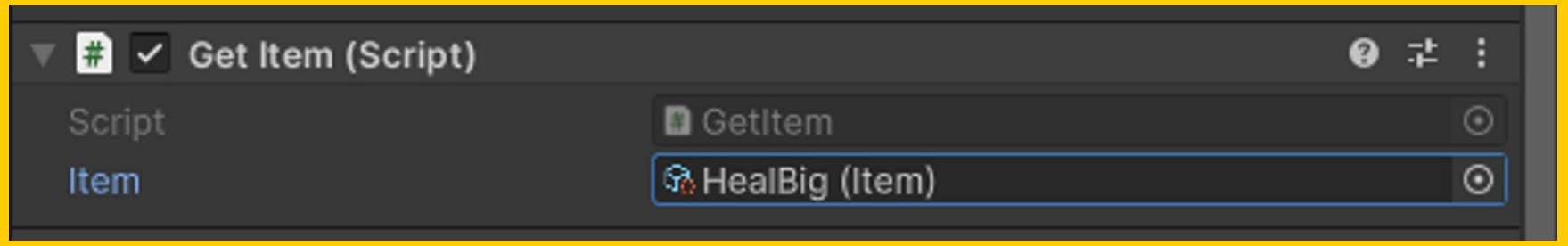


Item

각 젤리들의 정보가 담겨 있는 ScriptableObject 생성



Player가 해당 정보를 읽고 아이템 발동!



Item

[Dash]
- DashState 출력
- Player스크립트의 MapScrollSpeed를 일시적으로 높여 빠른 달리기 효과 연출

[Gigantic]
- Player의 localScale값을 일시적으로 확대

[Magnetic]
- 일정 시간동안 Player에 자성 발생
- Player 주변의 Jelly들을 끌어당겨 획득

[Basic Jelly & GomJelly]
- player의 Jelly포인트 증가

[Silver & Gold Coin]
- player의 Coin값 증가

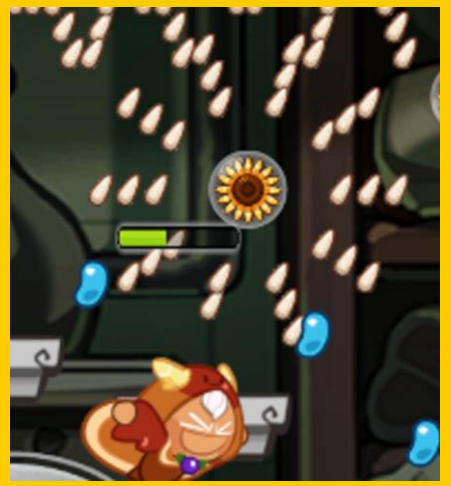
[BONUSTIME 젤리]
- enum값으로 게임 스크린 내 UI 변화

Item_Change Map

[Heal]
- Map1이 끝날 때즈음 획득 시 Map2로 넘어감
- 체력 +50 되는 기능

```
public void ChangeMaterial(Material[] mat_map)
{
    //Transform[] children = transform.GetComponentsInChildren<Transform>();
    Transform[] children = GameObject.Find("DS_Map").transform.GetComponentsInChildren<Transform>();

    foreach (Transform child in children)
    {
        if (child.gameObject.name == "BackGround1")
        {
            MeshRenderer meshRenderer = child.gameObject.GetComponent<MeshRenderer>();
            if (meshRenderer != null && mat_map != null) // mat_map.Length을 쓸 필요가 있는가?
            {
                meshRenderer.sharedMaterials = new Material[] { mat_map[0] };
            }
        }
        if (child.gameObject.name == "BackGround2")
        {
            MeshRenderer meshRenderer = child.gameObject.GetComponent<MeshRenderer>();
            if (meshRenderer != null && mat_map != null && mat_map.Length > 0)
            {
                meshRenderer.sharedMaterials = new Material[] { mat_map[1] };

                //BGM재생
                AudioClip bgmAudioClip = GameManager.Instance.LoadAudioClip(bgmAudioClipPath);
```

팬케이크맛 쿠키



- 기본쿠키로 구축해놓은 StatePattern을 그대로 적용
- 팬케이크맛 쿠키만의 특수 능력 스크립트만 따로 추가
  - ㄴ 도토리 쿠키 생성: 20개의 도토리 쿠키 획득 시 해바라기 젤리 생성.
  - ㄴ 해바라기 젤리를 중심으로 꽃잎을 연상시키는 씨앗 젤리가 생성.

달빛술사 쿠키

- 기본쿠키로 구축해놓은 StatePattern을 그대로 적용
- 달빛술사 쿠키만의 특수 능력 스크립트만 따로 추가
  ㄴ Star 젤리 생성: 20개의 젤리 획득 시 Star가 날아가 장애물을 없앤다.

파티클 시스템

04. 게임 시연

THANK YOU