

ItemSystem 説明書

1. Systemへの接続

まず、PlayerクラスにIPlayer2ItemSystemインタフェースを追加して欠損した属性を補足しましょう。

唯一改修すべき属性は以下です、ほかのはそのままで大丈夫。

```
//糸持っていますか  
🔥 经常调用 📦 1用法  
public bool HadSilk { get; }
```

このようになっていますね。

```
📦 0+1用法  
public bool HadSilk =>  
    mSilkData.SilkCount > 0;
```

ここまで、接続が完成しました。

2. 使用方法

プレイヤーのほう

例：

```
if (Input.GetKeyDown(KeyCode.A))
{
    this.UseItem(player: this);
}
```

ItemManagerのほう

道具生成

```
//道具生成
ItemSystem.Instance.SpawnItem(Vector3.zero);
```

道具をプレイヤーにあげる、例：

```
TypeEventSystem.Instance.Send(e: new PlayerGetItem()
{
    player = other.gameObject.GetComponent<IPlayer2ItemSystem>(),
});
```

必ず先接続したのIPlayer2ItemSystemインタフェースを引数としてEventで転送してください。そのあと、Systemが自動的に道具分配の仕事をやりますので、気にならずでも構いません。

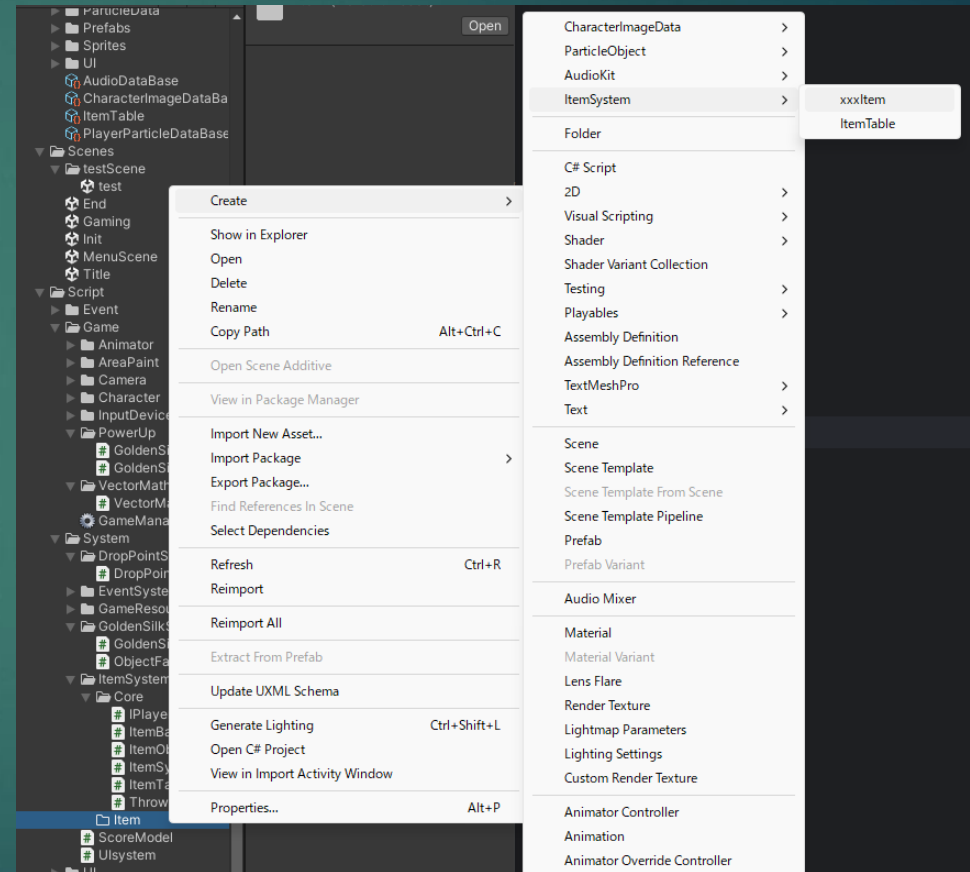
3. 道具の作り方

道具は自体作用と投げる二つ分けられます。

まずは自体道具、例：

```
[CreateAssetMenu(menuName = "ItemSystem/xxxItem")]  
无资源用法  
public class ExampleItem : ItemBase  
{  
    0+3 用法  
    public override void OnUse(Player player)  
    {  
        base.OnUse(player);  
        //ここからItemのロジックコード書いてください  
        Debug.Log(message: "Hello world!");  
    }  
}
```

Itemファイルの中に道具のAssetObj生成します。



毎道具が自分の独立のクラスが必要になりますのでご注意ください。一個つづ作っていきましょう。

3. 道具の作り方

投げるアイテムは少し特別です。

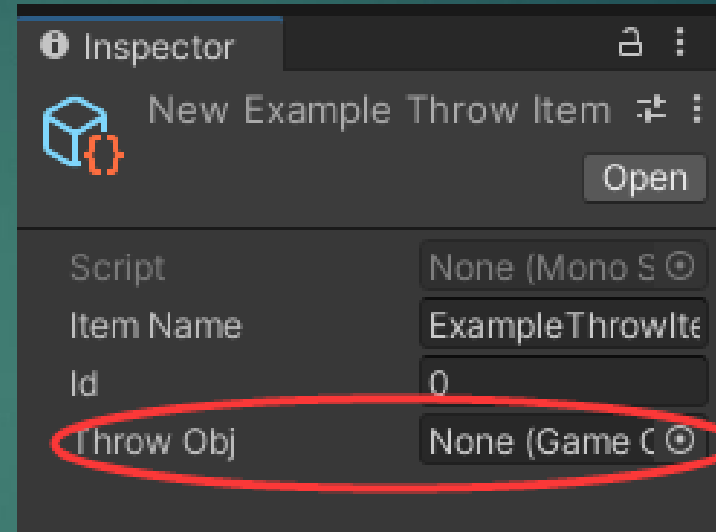
```
[CreateAssetMenu(menuName = "ItemSystem/xxxxxItem")]
//无资源用法
public class ExampleThrowItem : ThrowItem
{
    //何もしなくていい
}
```

ロジックコードは自分で作成したのprefabのコントローラみたいなscriptに書いていきます。

```
/// <summary>
/// 投げるアイテム
/// </summary>
//无资源用法
public class ThrowItem : ItemBase
{
    //投げるItemのprefab
    public GameObject ThrowObj; //未更改

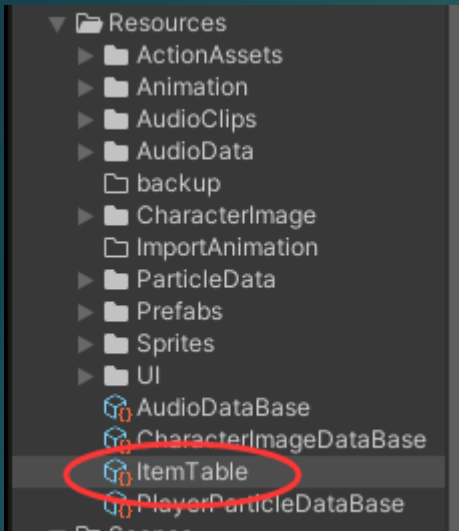
    //投げるItemには、一般的このOnUseメソッドをoverrideの必要がない
    //0+3 用法
    public override void OnUse(Player player)
    {
        base.OnUse(player);
        Instantiate(ThrowObj, player.transform.position,
            (Quaternion) rotation: quaternion.identity);
    }
}
```

作りましたprefabをここに持ち込んでください。



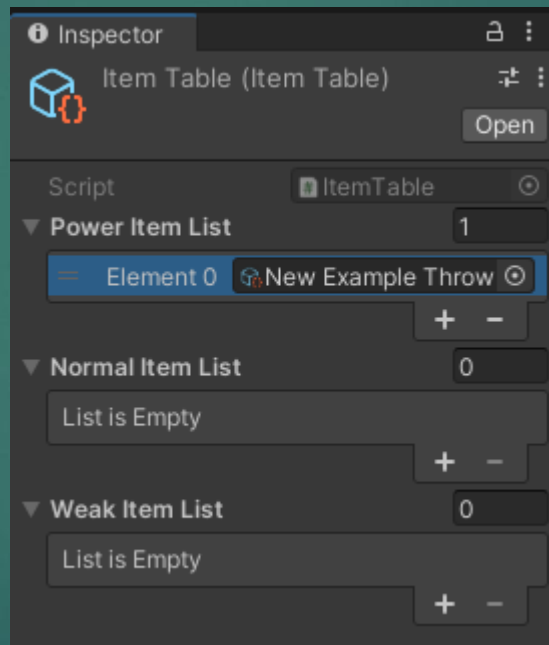
4. ItemTableに持ち込み

Tableの場所はこちらです



ここに、さき作成したのItemをプランナーの指示に従って、対応のリストに入れてください。

例：



やっと終わりましたね、お疲れ様でした。