ItemDetails

ID

物品详细数据

public struct InventoryItem

{

public int itemID;

public int itemAmount;

}

物品数量 物品ID

public class AnimatorType

{

public PartType partType;

public PartName partName;

public AnimatorOverrideController overrideController;

}

动画类型 动画名字 动画控制器

2级 类

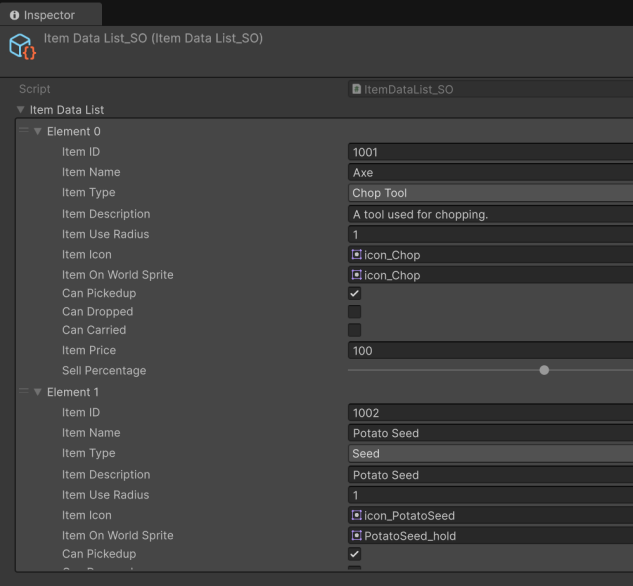
public class ItemDataList\_SO : ScriptableObject

{

public List<ItemDetails> itemDataList;

}

存储数据列表



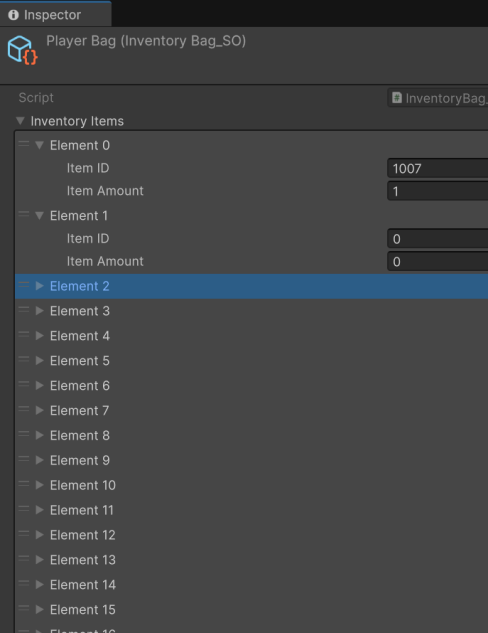
public class InventoryBag\_SO : ScriptableObject

{

public List<InventoryItem> inventoryItems;

}

ID 数量 背包数据



public class Item : MonoBehaviour

{

public int itemID;

private SpriteRenderer spriteRenderer;

public ItemDetails itemDetails;

private BoxCollider2D boxCollider2D;

public void Init(int ID)

{

Vector2 newSize = new Vector2(spriteRenderer.sprite.bounds.size.x, spriteRenderer.sprite.bounds.size.y);

boxCollider2D.size = newSize;

boxCollider2D.offset = new Vector2(0, spriteRenderer.sprite.bounds.center.y);

}

}创建一个prefab，生成物品的时候自动生成

public class ItemPickup : MonoBehaviour

{

private void OnTriggerEnter2D(Collider2D other)

{

Item item=other.GetComponent<Item>();

if (item != null)

{

if (item.itemDetails.canPickedup)

{

InventoryManager.Instance.AddItem(item, item.itemDetails.canPickedup);

}

}

}

}

public class ItemToolTip : MonoBehaviour

public void SetupTooltip(ItemDetails itemDetails, SlotType slotType)

显示说明栏

3级

public class InventoryManager : SingletonMonoBehaviour<InventoryManager>

{

[Header("item data")]

public ItemDataList\_SO itemDataList\_SO;

[Header("inventory data")]

public InventoryBag\_SO playerBag;

public ItemDetails GetItemDetails(int ID)

使用ID返回详细数据

public void AddItem(Item item, bool toDestroy)

To do

private int GetItemIdexInBag(int itemID)

Todo