

# 背包/仓库系统

重点部分：

1. 背包/仓库格子交互逻辑
  - a. 增：添加物品到格子中
  - b. 删：从格子中移除物品
  - c. 改：更改格子中的物品
2. 背包/仓库数据更新存储逻辑

核心脚本：

1. InventoryManager.cs
  2. UI\_MainInventoryWindow.cs
  3. UI\_ItemSlot.cs
- 

## 主要功能

初始化

1. 从存档中读入保存的数据
2. 初始化每个格子 and 对应父窗口的信息（详情见下方<格子初始化>）
3. 如果有武器则将武器模型更新到角色模型身上

获取某个物品数量

1. 根据物品对应的configId变量当前背包/仓库格子，如果当前格子非空并且物品id一直则加上物品数量
2. 返回物品数量累计结果

## 获取空格子

1. 遍历当前背包/仓库格子，如果当前格子为空则返回对应的索引，否则返回-1代表背包/仓库满了

## 设置格子中的内容

1. 将传入的物品数据设置到传入索引对应的背包/仓库格子数据里
2. 调用格子初始化方法更新UI，详情见<格子初始化>

## 移除格子中的内容

1. 将传入的索引对应的格子数据设置为null
2. 调用格子初始化方法将UI更新为默认状态，详情见<格子初始化>

## 检查和添加物品到空格子

1. 得到空格子的索引，详情见<获取空格子>
2. 如果当前索引合法则在对应索引处创建一个物品，详情见<设置格子中的内容>

## 添加物品

1. 根据传入的物品configId获取物品配置

---

## 格子初始化

1. **Init** 设置格子索引值、对应的父窗口（可能是仓库或者物品快捷栏）和格子的UI背景
2. **InitData** 通过传入的格子数据更新当前格子UI效果，空格子时恢复默认UI效果，格子中有物品时打开物品图标并设置不同的数值效果，例如武器是百分比，消耗品和材料都是个数

---

常见问题：通过常见问题串联起模块执行顺序

1. 玩家捡到物品后如何添加到背包/仓库里？
  2. 玩家扔掉物品后背包/仓库里的数据应该如何更新？
  3. 玩家交换物品时的逻辑是什么？
  4. 玩家使用物品时的逻辑是什么？
  5. 游戏结束后背包/仓库中的数据怎么保存？
  6. 继续游戏时如何恢复背包/仓库中的数据？
- 

背包/仓库数据更新存储逻辑

1. 什么状态下触发背包/仓库的存储机制？

---

数据结构

```
// 通用物品栏格子
[Serializable]
public class InventoryData
{
    // 物品栏中装的物品
    public ItemData[] itemDatas { get; protected set; }

    public InventoryData(int itemCount) {
        itemDatas = new ItemData[itemCount];
    }

    // 移除某一个物品
    public void RemoveItem(int index) {
        itemDatas[index] = null;
    }
}
```

```

        // 放置某一个物品
        public void SetItem(int index, ItemData itemData) {
            itemDatas[index] = itemData;
        }
    }

    // 物品快捷栏数据
    [Serializable]
    public class MainInventoryData : InventoryData
    {
        public ItemData weaponSlotItemData { get; private set; }

        // 物品快捷栏构造函数
        public MainInventoryData(int itemCount) : base(itemCount) {}

        // 移除武器
        public void RemoveWeaponItem() {
            weaponSlotItemData = null;
        }

        // 放置武器
        public void SetWeaponItem(ItemData itemData) {
            weaponSlotItemData = itemData;
        }
    }
}

```