



篝火

重点部分：

1. 篝火燃料如何添加
2. 篝火如何制作食物
3. 篝火光照随燃料剩余量变化

核心脚本：

1. Campfire_Conroller.cs
2. CampfireConfig.cs
3. CampfireData.cs



篝火管理器 Campfire_Conroller.cs

说明：篝火在刚开始建造时会有一定燃料，随时时间变化燃料会一点点消耗，燃料越多火焰越明亮。篝火在预览模式中是不显示火焰效果的，只有建造后才有火焰效果。

重要属性：

```
[SerializeField] new Light light;           // 灯光组件:点光源
[SerializeField] GameObject fire;           // 火焰组件
[SerializeField] private AudioSource audioSource; // 火焰音效
private CampfireConfig campfireConfig;     // 篝火配置
private CampfireData campfireData;        // 篝火数据
private bool isOnGround;                   // 篝火是否已建造
```

初始化方法

1. 调用基类初始化方法记录当前篝火所在地图块和篝火mapObjectId，并解锁相应科技树id
2. 从配置管理器中获取对应的篝火配置
3. 如果当前篝火是从存档中恢复则从存档数据中加载，否则需要创建一份新的篝火数据并设置默认燃料值后添加到存档中
4. 根据当前篝火燃料<设置光照>并记录当前篝火已经建造完毕(isOnGround=true)

Update

1. 如果游戏已经初始化完毕并且篝火已经建造完毕
 - a. <更新篝火燃料>

更新篝火燃料

1. 如果当前篝火燃料为0则代表篝火已经熄灭
2. 否则使用当前篝火燃料剩余量-时间*篝火燃料消耗速度计算得到当前帧篝火剩余燃料

3. 根据当前篝火燃料<设置光照>

预览模式

1. 执行基类预览方法
2. 设置当前篝火状态为建造中(isOnGround=false)
3. 关闭灯光和火焰效果 `SetLight(0);`

设置光照

说明：篝火剩余燃料可以影响光照强度、范围和篝火音量

1. 通过篝火剩余燃料设置当前灯光、篝火音效组件是否打开
2. 如果当前剩余燃料>0
 - a. 根据传入的燃料值计算当前剩余燃料比例
 - b. 根据燃料比例设置灯光强度light.intensity
 - c. 根据燃料比例设置光照范围light.range
 - d. 设置篝火音量是当前剩余燃料比例

鼠标拖拽方法

1. 根据传入的itemId判断当前物体是否在可燃烧物品配置中，如果在则取出对应燃料值
 - a. 如果在配置里则更新当前篝火数据campfireData.currentFeulValue+燃料值
 - b. 根据新的燃料值<设置光照>
2. 根据传入的itemId判断当前物体是否在可烘焙食物配置中，如果在则取出对应的食物id
 - a. 检查当前篝火是否熄灭，如果熄灭则提醒玩家需要点火后才能烘焙
 - b. 调用背包/仓库管理器根据食物id向背包中添加一个烘焙后的食物

篝火配置 CampfireConfig.cs

```
[CreateAssetMenu(fileName = "篝火配置", menuName = "Config/篝火配置")]
public class CampfireConfig : ConfigBase
{
    [LabelText("默认燃料数值")]
    public float defaultFuelValue;
    [LabelText("最大燃料数值")]
    public float maxFuelValue;
    [LabelText("燃烧速度/每秒消耗量")]
    public float burningSpeed;
    [LabelText("最大灯光强度")]
    public float maxLightIntensity;
    [LabelText("最大灯光范围")]
    public float maxLightRange;
    [LabelText("燃料和物品对照关系")]
    public Dictionary<int, float> itemFuelDict;
    [LabelText("可以烧烤的物品对照关系")]
    public Dictionary<int, int> bakedItemDict;

    public bool TryGetFuelValueByItemId(int itemId, out float fuelValue) {
        return itemFuelDict.TryGetValue(itemId, out fuelValue);
    }

    public bool TryGetBakedItemByItemId(int itemId, out int bakedItemId) {
        return bakedItemDict.TryGetValue(itemId, out bakedItemId);
    }
}
```

篝火数据 CampfireData.cs

```
[Serializable]
public class CampfireData : IMapObjectTypeData
{
    public float currentFeulValue;
}
```