



输入系统

重点部分：

1. 鼠标是否点击到了某些特定的游戏对象上
2. 鼠标点击地图对象是否可以互动

核心脚本：

1. InputManager
-

初始化

1. 设置输入检查标志 `wantCheck`，由于在建造系统中进入建造状态时需要关闭鼠标与地图对象的交互功能，因此需要单独设置一个标志位进行检查

Update

1. 检查游戏场景是否初始化完毕，当游戏场景彻底初始化完毕后可以正常使用输入系统
2. <检查鼠标选中地图对象是否可以互动>

检查鼠标是否在UI上

1. 创建 `PointerEventData` 并将position设置为当前鼠标位置 `Input.position`
2. 使用 `EventSystem` 中的检测方法 `RaycastAll` 进行检查UI
 - a. 遍历所有的检测结果，如果当前结果是UI但是不是Mask模块则返回true
 - b. 如果遍历完所有结果并未找到

检查鼠标是否在大型地图对象上

1. 使用 `Physics.Raycast(Camera.main.ScreenPointToRay(Input.mousePosition), 1000, bigMapObjectLayer)` 去检测鼠标点击位置是否出现大型地图对象 (`Layer == "bigMapObjectLayer"`)

获取鼠标在地面上的世界坐标

1. 如果通过射线检测的方式能够在 `GroundLayer` 上检测到, 则返回鼠标位置作为世界坐标
2. 否则返回 `vector3.zero`

检查当停止拖拽格子上的物品时是否点击到了建筑物身上

1. 如果通过射线检测的方式能够在 `BuildingLayer` 上检测到, 则认为鼠标点击到了建筑物
 - a. 考虑到不同建筑物可能有不同的处理方法, 所以执行基类 `BuildingBase` 中的结束拖拽的逻辑 `OnSlotEndDragSelect`

检查鼠标选中地图对象是否可以进行互动

1. 如果检查标志 `wantCheck==false` 则退出
2. 仅当鼠标按下时执行接下来的逻辑 (一直按下/单独点击鼠标)
 - a. `GetMouseButton`: 每一帧鼠标处于按下状态都会返回true
 - b. `GetMouseButtonDown`: 鼠标按下的那一帧返回true
3. <检查鼠标是否在UI上>, 如果鼠标在UI上则无法进行互动
4. 使用射线检查当前点击对象是否未非建筑物类 (例如地面掉落物品、AI、树木石头等地图对象)
 - a. 将数据 `hitInfo` 发送给玩家控制器进行处理
`Player_Controller.Instance.OnSelectMapObject(hitInfo, mouseButtonDown);` , 该方法中会判断玩家能否拾取/攻击/采集该地图对象
5. 检查鼠标是否在建筑物上, 如果鼠标点击了建筑物需要查看当前建筑物是否能进行互动
 - a. 互动要求: 1. 玩家与建筑物之间距离小于配置阈值

- b. 执行建筑物子类 `OnSelect()` 方法，例如储物箱的该方法将会打开箱子展示储物箱 UI