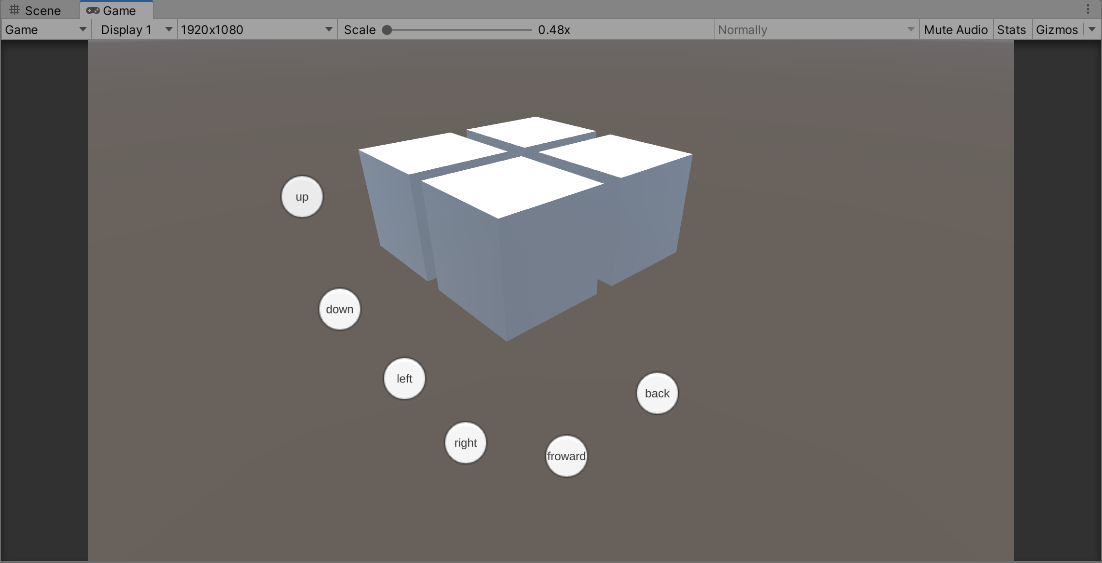


# 旋转



2024年12月16日21点28分

当前状态：六个按钮可以分别设置显示对应的面的对象。

现在有一个问题，如何将6个按钮关联起来。

2024年12月17日22点08分：

按钮已成功关联

旋转已完成

问题：旋转后当需要确定对应关系时，存在无法通过位置信息找到对应对象的情况。

2024年12月18日20点19分：

//return a <= b

public static bool Less(Vector3 a, Vector3 b , float e = 0.01f)

{

if ((a.x - b.x)>e)

{

return false;

}

else if ((a.y - b.y) > e )

{

return false;

}

else if ((a.z - b.z) > e )

{

return false;

}

return true;

}

由于旋转时，不一定准确地旋转90°，导致最后的位置不确定。

问题出在两个地方，

第一个是上述的判定存在问题，之前使用的是if(a.x > b.x)的形式，但是存在期望坐标中的某可能为0.9999999，而另一个坐标中存在与之相对的值为1.0000001的情况，导致无法通过判定；

第二个是字典，字典值用的是vector3，同样由于上述原因导致无法定位到指定值的情况，最初使用了如下的比较类来进行误差判定，但是由于没有定位到问题一，而使用了List的方案进行大致判定。

public class Vector3EqualityComparer : IEqualityComparer<Vector3>

{

private readonly float \_epsilon;

public Vector3EqualityComparer(float vIn\_Epsilon)

{

\_epsilon = vIn\_Epsilon;

}

public bool Equals(Vector3 x, Vector3 y)

{

return (Mathf.Abs(x.x - y.x) < \_epsilon) &&

(Mathf.Abs(x.y - y.y) < \_epsilon) &&

(Mathf.Abs(x.z - y.z) < \_epsilon);

}

public int GetHashCode(Vector3 obj)

{

return (obj.x / \_epsilon).GetHashCode() ^ (obj.y / \_epsilon).GetHashCode() ^ (obj.z / \_epsilon).GetHashCode();

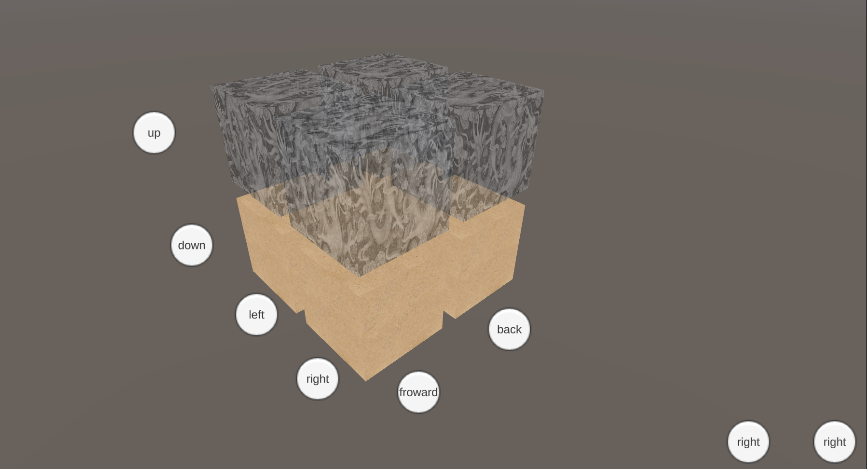
}

}

原因找到，后续再考虑使用Physical.Overlapbox以及链表再做两版，对比其性能。

着手进行透明shader的编写。

2024年12月18日22点20分：



现在已具备透明效果，但看起来感官不好，等梦境迷钟实现之后再具体参考铁的玩法进行修改。铁的箱子应该具备自发光、bloom、透明效果，目前仅实现了透明效果TODO

明天尝试把完成逻辑编写完毕

2024年12月19日星期四21点14分

基础玩法完工，最后一步的检测偷懒了，用了unity的API。不过好歹能看了，待会儿把场景添加到Fork场景的按钮事件中，尽快完成梦境迷钟。

添加XML配置文件，用于编写关卡信息

ERROR：在手机上由于帧率被限制在30，导致旋转后的位置与期望值差异过大，导致重新定位出现异常

2024年12月21日13点36分 ： 已解决重新定位问题