# Unity查缺补漏

1. **Image.fillAmount(): 1表示完全填充，0表示不填充**
2. **参数面板默认是欧拉角选择，使用会产生的死锁，进而跳变。**

**当你需要做动画的时候，你想从（90，0，0）插值到（150，90，90）的时候，写出了这样的代码**

**transform.eulerAngles = Vector3.Lerp(transform.eulerAngles, new Vector3(150, 90, 90)**

**因为y和z会突然跳变，就会形成死锁，甚至卡住电脑。那为什么这个问题现在还存在呢因为Unity设置x在中间就是因为一般x不会旋转90度的，常见的旋转插值是对Camera进行的，而如果绕着X轴旋转90度，则意味着正向上，或是正向下，这两种情况都是非常少见的。**

1. **四元数：数据的范围是-1 - 1，每次旋转需要两个四元数**

**例如Quaternion.AngleAxis(float angle, Vector3 axis)，它可以返回一个绕轴线axis旋转angle角度的四元数变换。我们可以一个Vector3和它进行左乘，就将得到旋转后的Vector3。在Unity里只需要用一个“ \* ”操作符就可以进行四元数对向量的变换操作。如果我们想要进行多个旋转变换，只需要左乘其他四元数变换即可。**[【Unity】旋转的尽头是使用四元数让物体旋转\_unity 四元数 旋转-CSDN博客](https://blog.csdn.net/leoysq/article/details/136737304)

1. **lambda表达式**

**单参数：x=>x\*x // 传入参数x，返回x\*x**

**多参数：（x,u）=>x == u // 传入参数x，u，返回bool值**

1. **反射优化switch**
2. **Aa**
3. **A**
4. **Aaa**
5. **Aaa**
6. **A**
7. **A**
8. **A**
9. **Aa**
10. **A**
11. **A**
12. **A**
13. **d**