

### 6.2.1 基础战斗框架编写

通常伤害、属性状态等信息存放在战斗信息的结构体里，在触发动画事件后将战斗信息发送给受攻击者。我们可以结合 Unity 的碰撞事件，在产生碰撞后获取相应的战斗信息，而不是主动将战斗信息发送给对方，这对于后期受击判定调试有一定的帮助。如图 6.9 所示，左侧为传统的直接发送攻击信息的方式，右侧为碰撞时被动获取攻击信息的方式。

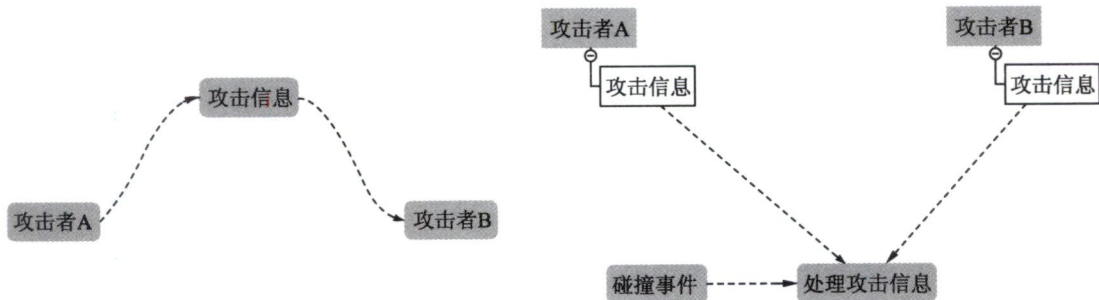


图 6.9 战斗状态传递示意图

(1) 这里把战斗对象的基础类叫作 **BattleObject**，它是继承于 **MonoBehaviour** 的。

```

public class BattleObject : MonoBehaviour
{
    void OnTriggerEnter(Collider collider)           //碰撞触发
    {
        var otherBattleObject = collider.transform.GetComponent<BattleObject>();
        //过滤掉没有挂载 BattleObject 的对象
        if (otherBattleObject == null) return;
        //具体逻辑操作（暂时先省略）
    }
}

```

(2) 接下来为战斗对象 (**BattleObject**) 加入阵营与战斗回调事件，不同阵营 (**Faction**) 的单位会收到伤害信息，对于更复杂的情况还可以继续针对阵营增加二维表等，这里只做简单的布尔判断，并且只针对玩家和敌人阵营。首先加入阵营常量表：

```

public static class EasyFactionConst
{
    public const int PLAYER = 1;           //玩家
    public const int ENEMY = 2;           //敌人
}

```

(3) 然后对之前的脚本进行扩展：

```

public class BattleObject : MonoBehaviour
{
    public int faction = EasyFactionConst.PLAYER;
}

```