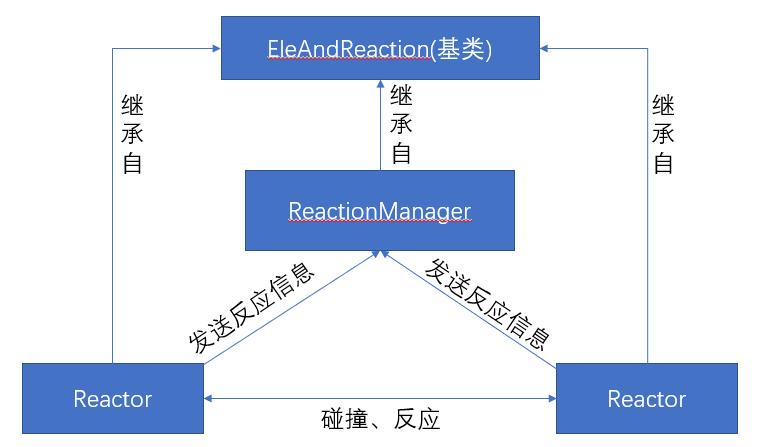
本次更新了元素反应的代码框架

接下来是系统与代码介绍



代码结构如图。

游戏中的每一个可参与元素反应的物体上都会挂一个Reactor，该脚本将监测碰撞，并将合法反应打包成反应信息包“ReactionMsg”，并发送给ReactionManager进行下一步处理。在该过程中，也就是游戏中的任何一个反应过程中，都只能由两个Reactor参与，而两者都会向manager发送反应信息。Manager会自己清除重复的反应信息。

Reactor与Manager都继承自EleAndReaction基类，该基类中包含了元素种类Enum，以及各元素反应函数，这些函数在Manager中被重写，然后根据需要被执行。

Manager将接收并处理所有的元素反应请求。

Reactor中包含了销毁函数，元素的destroy过程各不相同，需求复杂，要销毁一个元素应调用Reactor中的函数。要实现定制化的销毁需求，须重写脚本，继承Reactor，重写销毁函数。

元素系统的补充解释：

可参与反应的物体将被分为三类，强元素、弱元素以及角色（包括玩家与AI）。

强元素参与反应后不发生变化，如水洼；弱元素反应后销毁，如子弹；只有角色才可以沾染元素。

在这里介绍一下水面结冰与水体导电的处理方法：

在水面生成一层不与水接触的冰/电，这层东西视作弱元素，可参与反应消失，存在一段时间后也会消失；这种情况，就应新开脚本，继承自Reactor，根据需要写补充代码。

目前的代码设计暂时没有考虑过多网络联机相关的问题。