主要针对帧同步

先看这两个帖子

[帧锁定同步算法 - Skywind Inside](https://skywind.me/blog/archives/131)

[同步算法-帧同步 - 独一无二~ - 博客园](https://www.cnblogs.com/zhaobangyu/articles/18259335)

理解了帧同步本质上是个回合制，本地就是个播放器之后，收到服务器的播放要求后就会暂停或者快进到某一帧。

之后可以看看这个视频

[TSRPC + Cocos，多人实时对战 So Easy！原来帧同步和状态同步可以并用，200ms网络延迟也能实现零延迟操作体验，50分钟把多人实时对战说透了\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV16F411z7iP/?spm_id_from=333.337.search-card.all.click&vd_source=2fd56c06057ed87c77543a25d43c6a6e)

根据上面的视频内容，我写了个测试脚本

test\_network\_frame.py

简单说下这个脚本的处理流程：

1. server和client 都是跑在一定的逻辑帧率（比如60fps），然后server每隔几帧就广播一次所有client的输入
2. client在逻辑tick过程中，只要有输入，就发送给server，server在准备广播的时候，去应用这些输入并且广播出去，这些输入就是经过server验证的输入，叫权威输入，并且server还会发送当前自己的state的checksum。而client在发送自己的输入的同时，也会将这个输入直接应用在自己本地（这个输入未经过服务器验证，所以叫预测输入）。
3. client收到server的state checksum（client收到的一般是之前state的checksum），如果这个state checksum和client对应的那个state的checksum不相等，说明client的预测是错误的，那么client就回滚，回滚之后再应用预测输入到自己当前的帧，所以client上的显示其实都是预测的输入，你会从画面上看到，无论client领先服务器多少，最后大家都会变成一致
4. 主要的问题就是，当client的输入发送给server的时候，已经超过了这个输入所在的turn，那么这个输入到底算哪一帧？这里我们做的是他的输入如果超过了turn，则这个输入相当于延迟输入了，也就是要改变帧号了，可以用server当前的帧号来当新的帧号。

当然还有影子跟随算法

[影子跟随算法（2007年老文一篇） - Skywind Inside](https://skywind.me/blog/archives/1145)

更多的项目理解

[游戏网络同步绿皮书 - 知乎](https://zhuanlan.zhihu.com/p/565875153)