

手机格斗网游该如何避免延迟？ - GameRes游资网



GameRes游资网授权发布 文 / [韦易笑](#)

早期 RTS，XBOX360 LIVE游戏常用同步策略是什么？格斗游戏多人联机如何保证流畅性和一致性？如何才能像单机游戏一样编写网游？

算法概念

该算法普遍要求网速RTT要在100ms以内，一般人数不超过8人，在这样的情况下，可以像单机游戏一样编写网络游戏。所有客户端任意时刻逻辑都是统一的，缺点是一个人卡机，所有人等待。

- 1.客户端定时(比如每五帧)上传控制信息。
- 2.服务器收到所有控制信息后广播给所有客户。
- 3.客户端用服务器发来的更新消息中的控制信息进行游戏。
- 4.如果客户端进行到下一个关键帧(5帧后)时没有收到服务器的更新消息则等待。

5.如果客户端进行到下一个关键帧时已经接收到了服务器的更新消息，则将上面的数据用于游戏，并采集当前鼠标键盘输入发送给服务器，同时继续进行下去。

- 6.服务端采集到所有数据后再次发送下一个关键帧更新消息。

这个等待关键帧更新数据的过程称为“帧锁定”

应用案例：大部分RTS游戏，街霸II(xbox360)，Callus模拟器。

算法流程

客户端逻辑：

判断当前帧是否是关键帧K1：如果不是跳转(7)。

如果是关键帧，则察看有没有K1的UPDATE数据，如果没有的话重复2等待。

采集当前K1的输入作为CTRL数据与K1编号一起发送给服务器

从UPDATE K1中得到下一个关键帧的号码K2以及到下一个关键帧之间的输入数据。

从这个关键帧到下一个关键帧K2之间的虚拟输入都用。

令 $K1 = K2$ 。

执行该帧逻辑：

跳转(1)

服务端逻辑：

收集所有客户端本关键帧K1的CTRL数据(Ctrl-K)等待知道收集完成所有的CTRL-K。

根据所有CTRL-K，计算下一个关键帧K2的Update，计算再下一个关键帧的编号K3。

将Update发送给所有客户端

令 $K1=K2$

跳转(1)

服务器根据所有客户端的最大RTT，平滑计算下一个关键帧的编号，让延迟根据网络情况自动调整。

算法演示

我根据该算法将街机模拟器修改出了一个可用于多人对战的版本，早期有一个叫做kaillera的东西，可以帮助模拟器实现多人联机，但是并没有作帧锁定，只是简单将键盘消息进行收集广播而已，后来Capcom在PSP和360上都出过街霸的联网版本，但是联网效果不理想。这个算法其实局域网有细就经常使用了，只是近年来公网速度提高，很容易找到RTT<50ms的服务器，因此根据上述算法，在平均RTT=100ms(操作灵敏度1/10秒)，情况下，保证自动计算关键帧适应各种网络条件后，就能够像编写单机游戏一样开发网游，而不需状态上作复杂的位置/状态同步。

从上图的演示中可以看到，两个模拟器进程都在运行1941这个游戏，两边客户端使用了该算法，将逻辑统一在一个整体中。

最后这张图是运行KOF99的效果图，两边完美同步，上图是我开发的街机对战效果，在公网环境下，只要不是小型宽带用户，一般电

信或者联通用户都能有一个比较好的体验。

帧间无等待改进

针对传统帧锁定算法网速慢会卡到网速快的问题，实践中线上动作游戏通常用“定时不等待”的方式再每次Interval时钟发生时固定将操作广播给所有用户，不依赖具体每个玩家是否有操作更新：

单个用户当前键盘上下左右攻击跳跃是否按下用一个32位整数描述，服务端描述一局游戏中最多8玩家的键盘操作为：int player_keyboards[8]；

服务端每秒钟50次向所有客户端发送更新消息(包含所有客户端的操作和递增的帧号)：update=(FrameID, player_keyboards)

客户端就像播放游戏录像一样不停的播放这些包含每帧所有玩家操作的update消息。

客户端如果没有update数据了，就必须等待，直到有数据到来。

客户端如果一下子收到很多连续的update，则快进播放。

客户端只有按键按下或者放开，就会发送消息给服务端(而不是到每帧开始才采集键盘)，消息只包含一个整数。服务端收到以后，改写player_keyboards

虽然网速慢的玩家网络一卡，可能就被网速快的玩家给秒了(其他游戏也差不多)。但是网速慢的玩家不会卡到快的玩家，只会感觉自己操作延迟而已。另一个侧面来说，土豪的网宿一般比较快，我们要照顾。

随机数需要服务端提前将种子发给各个客户端，各个客户端算逻辑时用该种子生成随机数，另外该例子以键盘操作为例，实际可以以更高级的操作为例，比如“正走向A点”，“正在攻击”等。帧锁定系列方法目前也成功的被应用到了若干线上实时动作游戏中。

关于帧锁定系列的方法有很多类似实现(包括后面提到的帧间无等待改进，包括 Lock Step等)，但是他们的核心都是一个：保证所有客户端每帧的输入都一样。这样的方式被格斗游戏，RTS和足球(FIFA类)、篮球(NBA)等体育和动作游戏大量使用，比如我们熟悉的各大战网平台游戏(Xbox Live等)，还有很多基于模拟器的街机对战平台。以及不少大型多人横版动作游戏。以开发便利，同步逻辑直观而受到大家欢迎。

而近两年动作游戏领域也涌现出一些新的改良方法，比如 Time Warp，以客户端先行+逻辑不一致时回滚的方式，带来了更好的同步效果，俗称时间回退。不果国内暂时没看到有游戏这么尝试，更多的是国外近两年的双人动作游戏比较多，要求游戏每帧状态都可以保存，逻辑上开发会复杂一些。国内大部分是超过两人出去副本的，在3-4人出去 PK的情况下，引入状态回退，会让整个效果大打折扣。不果2人的效果确实有所改进，有兴趣的同学可以搜索 Time Warp相关的论文。

2009年，云游戏(游戏远程渲染)技术得到广泛应用，客户端上传操作，服务端远程渲染，并以低延迟视频编码流的方式传回给客户端，用的就是这样类似的技术。客户端不需要高额的硬件，也不存在盗版问题，其中 Gaikai和 OnLive两家公司做的比较好。

2012年，Sony推出 Playstation Now技术，可以在 PSV和 PS3/PS4上玩云游戏，玩家不需要购买游戏就可以免费体验一定时间。使得 PSV/PS3等低端硬件也可以流畅的跑 PS4游戏。

但是目前国外网络环境下跑的还比较流畅，国内的网络环境要低延迟传送 HD画质的视频流还比较困难，视频都是比较费带宽的。但是帧锁定等保证每帧输入一致的算法，在当今的网络质量下传递一下玩家操作，还是没有任何问题的。