# 基本规则

* 客户端与服务端建立心跳机制,实时更新客户端的网络ping值.服务端和客户端同时记录
* 客户端建立发送消息队列,等待回应消息队列,丢弃消息队列.
* 设定客户端延迟最小精度单位,假定50ms.

# 基本行为

* 主动行为

1. **位置变更**

* 客户端收到用户移动指令后生成消息放入消息发送队列,客户端同时开始播放行为动画,同时对位置进行更新.
* 消息模块从发送队列提取消息,并且即时发送至服务端
* 发送成功

将消息放入等待回应列表,进入下一步骤.

* 发送失败

将消息放入丢弃消息队列,同时通知客户端该消息发送失败.客户端并不取消当前动画的播放和位置的更新. 消息模块通知逻辑模块消息发送失败,由逻辑模块处理后续的操作，流程终止.

* 等待服务端回应确认
* 正常返回

小于延迟最小精度单位的消息返回.基本上能和客户端的动作进行同步.将消息从等待消息队列删除.

* 延迟返回

客户端动作播放完毕,位置更新完毕后如果服务端还未返回的情况属于严重延迟.这种情况下客户端不需要撤销已经更新的本地位置.这个时候接收的消息需要客户端判断该消息是否客户端最后发送的同步消息.如果是的话客户端需要将角色的位置按照确认消息的数据进行更新(如果位置不一致则进行更新,如果一致则跳过).

1. **释放技能**

* 客户端收到用户释放技能指令后生成消息放入消息发送队列,客户端同时开始播放技能释放动画.
* 消息模块从发送队列提取消息,并且即时发送至服务端
* 发送成功

将消息放入等待回应列表,进入下一步骤.

* 发送失败

将消息放入丢弃消息队列,同时通知客户端该消息发送失败.客户端并不取消当前动画的播放.消息模块通知逻辑模块消息发送失败,由逻辑模块处理后续的操作,流程终止.

* 等待服务端回应确认
* 正常返回

小于延迟最小精度单位的消息返回.如果消息是确认消息客户度则播放对应的效果动画.如果有击中目标则显示和处理伤害/治疗数值.将消息从等待消息队列删除.

* 延迟返回

客户端技能动画播放完成后还未返回的情况下客户端同样不做任何处理.同时允许客户端继续发起其他操作.什么时候收到技能确认消息什么时候播放技能效果和进行技能数值逻辑处理.如果同时收到多个确认消息则客户端需要根据当前的情况顺序播放技能动画和显示处理技能数值.这个时候的播放需要根据延迟情况加快播放速度.

* 被动行为

1. **队友信息同步**

组队模式下,客户端收到服务端的队友信息同步(位置,释放技能)消息时.同步消息需要包含队友的移动速度,队友移动目标位置(如果是位置同步消息),队友释放技能，队友的客户端延迟,服务器处理消息时间等几个关键数据.

接下来计算队友的同步信息从队友客户端发出至服务端再广播至当前客户端这一个流程过来大概的总的网络延迟(队友客户端延迟 + 服务端处理时间 + 本地客户端延迟).

位置同步消息.根据队友速度和总的延时时间来预测队友当前可能已经移动到的位置.获取预测的位置后更新客户端队友显示.把队友当前的位置适当加速移动到预测位置,然后恢复队友的移动速度知道到达目的地.

动作同步消息,则客户端同样要根据总的延迟时间来加速队友的技能播放动画.

1. **被攻击**

网络上某一个客户端或者NPC（怪物）针对某个玩家（客户端）发起一个攻击请求，服务端接收到攻击请求后根据攻击方和被攻击方的属性计算攻击结果，然后将计算结果包含在攻击确认消息一同下发给被攻击.

被攻击方接收到攻击确认消息时提取服务端计算并且下发的攻击的结果信息并且更新屏幕，同时给服务端返回一条攻击完成消息.服务端收到攻击完成消息后更新被攻击客户端在服务端的状态，并且将刚才计算的攻击结果数据广播给攻击方（客户端），攻击方的队友（如果有组队），被攻击方的队友（如果有组队）.

在这个流程的过程中由于网络延迟的原因可能导致以下几种情况：

* 攻击确认消息发送失败

发生该种情况有可能被攻击方和服务端的网络连接已经断开，这个时候服务端不做任何逻辑处理，同时将该客户端网络断开的消息广播给可视区域里面所有的其他客户端玩家。

* 攻击完成消息接收延迟或接收超时

这种情况为服务端成功给被攻击方发送了攻击确认消息，但是客户端回应超时.这种情况下服务端依旧进行攻击结果数据的逻辑运算，并且放弃等待回应.同时将刚才计算的攻击结果数据广播给攻击方（客户端），攻击方的队友（如果有组队），被攻击方的队友（如果有组队）.广播完毕后给被攻击方发送一条状态更新消息，告知被攻击方受到攻击，并且造成多少伤害，扣除多少生命，同时将服务端保存的客户端状态一起下发，客户端收到更新消息则以服务端的数据为准进行自身的状态更新和修正.