**游戏流程：（主要写一下客户端流程）**

1. 连接
2. 收取并存储clientId和playerId(S2CSendClientId)
3. 选角色(C2SChooseChar)
4. 收取并存储playerId对应的charId(S2CSendCharId)
5. 收取并存储所有人playerId，charId(S2CAllCharId)
6. 根据playerId创建所有玩家
7. 选位置(C2SChooseLocation)
8. 收取并存储所有人位置(S2CAllLocation)
9. 接收开始消息(S2CGameStart)
10. 实时接收位置(S2CMove)
11. 客户端移动(C2SMove)

**网络包参数说明：**

S2CSendClientId

clientId 客户端id

playerId 玩家id

C2SChooseChar

charId 确定选哪个角色的角色id

S2CSendCharId

playerId 玩家id

charId 角色id

S2CAllCharId

playerId2CharId 所有人玩家id对应的角色id

C2SChooseLocation

playerId2CharId 所有人玩家id对应的角色id

x 地图数组的x值

z 地图数组的z值

S2CAllLocation

allLocation 所有人地图数组的x，z值

S2CGameStart

placeholder 占位符，无意义

S2CMove

allCharLocation 所有人在unity的坐标值

C2SMove

x 坐标x移动量

z 坐标z移动量

C2SAttack

weaponId 武器id

locationX 人物攻击时所在坐标x值

locationZ 人物攻击时所在坐标z值

S2CAttack

weaponId 武器id

x 泡泡应该放的地图数组x值

z 泡泡应该放的地图数组z值

S2CCellChange

mapId 地图id

x 变化的地图数组x值

z 变化的地图数组z值

value 变化后的值

S2CHPChange

playerId 玩家id

nowHP 变化后的血量

S2CDie

playerId 死亡的玩家id

S2CSendWinner

playerId 获胜的玩家id

S2CCellChange

mapId 地图id

x 地图数组x值

z 地图数组z值

value 地图变化后的值

**地图数组中值区间规则：**

0-1000表示可破坏格子的类型

2001-3000表示放在地上的可见道具，数值为道具id

10001-99999表示不可破坏的格子，不同地形不同id

100001-无穷 表示已经放在地上的将爆炸的泡泡，数值是一个泡泡的计数id，泡泡具体属性由服务器实例存储

原来的1001-2000表示玩家id，现在玩家id不再存放在地图数组中，玩家位置由服务器玩家实例存储，S2CMove中的x和z原来表示位移量，现在表示玩家所在的地图数组中的位置

隐藏在可破坏格子里的道具id由另一张相同大小的地图数组存储，相应位置存储道具id值

地图数组中的值一旦变化，会有S2CCellChange通知变化位置和新值，这样可能会和某些网络包重复，比如泡泡爆炸时有一个通知表示地图上这个位置值变成了0，同时泡泡爆炸的网络包会通知这个位置泡泡炸了，值变成了0，这里需要再考虑下是都通知还是只通知地图值变化后由客户端判断是什么变化

**道具系统**

道具增减

碰到道具即捡起道具

丢道具(丢到哪里由服务器控制)

使用道具

捡起道具：

S2CGetItem

playerId 玩家id

itemId 捡起物品的id

count 捡起物品的数量

客户端请求丢弃道具：

C2SDeleteItem

playerId 玩家id

itemId 捡起物品的id

count 捡起物品的数量

服务器丢弃道具：

S2CDeleteItem

x 丢弃位置数组x值

z 丢弃位置数组z值

playerId 玩家id

itemId 捡起物品的id

count 捡起物品的数量

客户端请求使用道具：

C2SUseItem

itemId 使用的道具id

服务器使用道具

S2CUseItem

playerId 使用道具的玩家id

itemId 使用的道具id