

# Clients/Server Communications

## (During the game)

### I – States

Here is a description of the different states.

#### a) *GS\_WAIT*

This is the state server and clients take when they are created. Server leaves it and takes the *GS\_INITGAME* state when two clients are connected. Then it sends *MSG\_INITGAME* message to clients. They turn into *GS\_INITGAME* state when they receive it.

Server ignores all messages from clients, except connection messages.

#### b) *GS\_INITGAME*

Server and clients take this state only once. Clients ask server to create teams and players with *MSG\_TEAMINFO* and *MSG\_PLAYERCREATE* messages. If server accepts, it creates the instance and sends an acknowledgment to clients. Then, they create the corresponding team or player.

When a client is ready, it sends the *MSG\_INITGAME* message back to the server. When both clients are ready, server turns into *GS\_INITKICKOFF* state.

#### c) *GS\_INITKICKOFF*

The server enters this state before a kick-off. It removes ball and players from field and sends a *MSG\_INITKICKOFF* message to the defender team.

The client enters this state too; it has to place its players. It sends a *MSG\_INITKICKOFF* message back to the server who checks the placement. If it is valid, it sends the message to the receiving team.

When both teams are correctly placed, server sends a *MSG\_INITKICKOFF* message again, to the kicking team; the client has to place the ball. When it is done, the server resolves the kick-off. If ball goes out of the field or in the defender zone, server turns into *GS\_TOUCHBACK* state and sends a *MSG\_GIVEBALL* message to the receiving team. If the ball finally lands in the attacker zone, game begins: server enters in *GS\_COACH1* or *GS\_COACH2* state and sends a *MSG\_NEWTURN* message.

#### d) *GS\_TOUCHBACK*

The ball has not landed in the correct zone. Client turns into this state when it receives a *MSG\_GIVEBALL* message. Then it sends it back with the id of the player to give the ball to.

If the player can carry the ball (on the field, standing, with hands...), the servers send the message back, as an acknowledgment. Then, servers enters in *GS\_COACH1* or *GS\_COACH2* state and sends a *MSG\_NEWTURN* message.

*e) GS\_COACH1/GS\_COACH2*

This is the turn of the corresponding coach. Server waits for actions from clients. It checks the client is allowed to realize it and send a MSG\_ILLEGAL message or resolve the action. If it is its turn, the client can declare or realize actions; else it can just chat or ask for an illegal procedure.

*f) GS\_REROLL*

The current team can use a team re-roll. When it has chosen, it sends a MSG\_REROLL message to the server. Then, server sends the message back as an acknowledgement and resolves the action

*g) GS\_BLOCK*

A client has to choose a block dice to resolve the block. When it has chosen, it sends a MSG\_BLOCKDICE or MSG\_REROLL message to the server. If the message is valid, it sends the message back and resolves the action.

The other client stays into GS\_COACH1 or GS\_COACH2 state.

*h) GS\_PUSH*

The client enters this state when it can choose a square to push a player, as a result of a block action. The server enters this state too. When it has chosen, it sends an ACT\_BLOCKPUSH message to the server. If the message is valid, it sends the message back and resolves the action.

The other client stays into GS\_COACH1 or GS\_COACH2 state.

*i) GS\_FOLLOW*

The client can choose if its player follows or not after a block. The server enters this state too. When it has chosen, it sends a MSG\_FOLLOW message to the server. If the message is valid, it sends the message back and resolves the action.

The other client stays into GS\_COACH1 or GS\_COACH2 state.

*j) GS\_SKILL*

A client can choose to use or not a skill. When it has chosen, it sends a MSG\_SKILL message to the server. If the message is valid, it sends the message back and resolves the action.

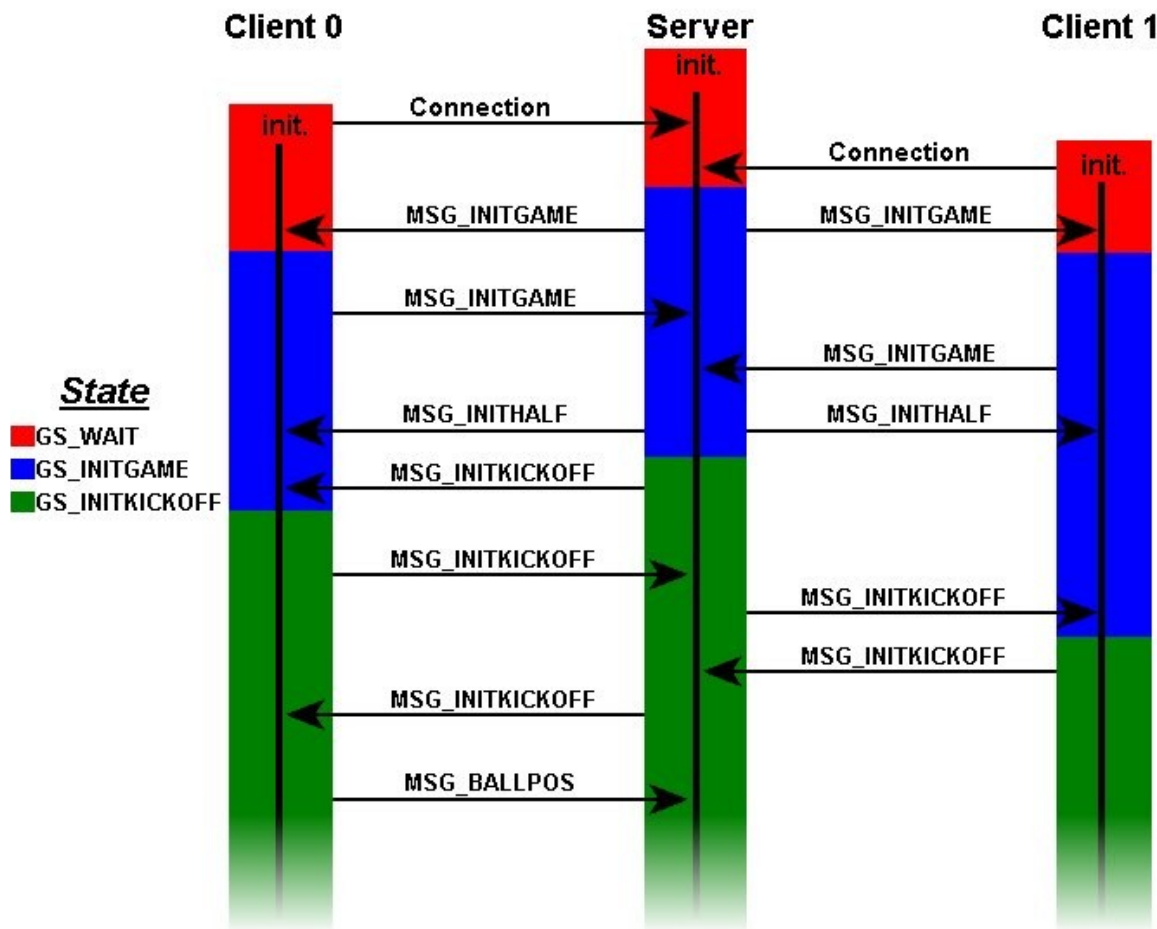
The other client stays into GS\_COACH1 or GS\_COACH2 state.

## II – Mechanisms

Here are explained some of the main mechanisms of the game.

### a) Initialization – global aspect

Initialization of the game uses many messages; it will be explained in different parts. First, server waits for clients. When both are connected, all needed instances are created (ball, players ...). Clients and server are in GS\_INITGAME state. Then, server initializes half-time and kick-off.



N.B.: Some messages (MSG\_TEAMINFO, MSG\_PLAYERPOS, ...) are not represented in this global scheme.