**Multiplayer handshaking** 1

**Host Player**

Flags blue, datagrams red

Loads a pre-start railway or multiplayer session

Multiplayer menu enabled

Selects 'Joins a session'

Completes boxes with host IP/Port/username

Red image

Clicks 'Send'

Yellow image

**Prepare for & begin session**

**Prepare for & begin session**

Host loads a pre-start railway

Multiplayer menu enabled

Selects 'Host a session'

Completes boxes with own IP & Port

Clicks 'Load coupling file'

File loads (or error message)

String grid loads all railways & sets

host's own railway + ready

Creates new InfoVector & loads

RailwayNames and RlyUserNumbers

("Host" as user for own railway)

Panel message 'Awaiting players'

PlayerMakingInitialContactFlag = true

All datagrams from player to host begin with a marker to define the type of data, the username (this

identifies the sender so the port and IP can be saved by the host after each contact - these can sometimes be changed by the router)

CouplingFileLoadedAwaitingPlayersFlag = true

**<-----------------------** Marker = 1+username+railwayname. Sent at 5 sec intervals until response received**-------------------------**

Checks correct railway & not already

allocated - returns message if not

Accept player? - rejection message if not

Accepts player, updates grid/InfoVector

End sequence, player can attempt to rejoin

**----------------------------** Error/rejection message **-------------------->**

Gives error/rejection message & cancels join

PlayerMakingInitialContactFlag = false

Each railway/user is allocated a number starting at 1 and increasing in railway name alphabetical order.

Stores own UserNumber

Compiles DynMapFromHost and DynMapToHost (blank service info)

Ready to begin button enabled

Green image & label changes - 'click ready to begin when ready'

**---------** Response: Send player usernumber+OneRailwayCouplingMap as buffer for the player**-------------->**

DynMapFrom & ToHost are std::maps with key a pair, 1st usernumber,

2nd a pair of H&V element locations, and value a class of service reference,

repeat number & time to exit in secs.

PlayerMakingInitialContactFlag = false

PlayerReadyToBeginFlag = true

Ready to begin button clicked

**<-------------------------------------------------------** Marker = 2+username+DynMapToHost as buffer**--------------------------------------**

Updates HostCombinedDynamicMap, updates grid & InfoVector

PlayerReadyToBeginFlag = false

PlayerAwaitingHostStartFlag = true

Clears join panel apart from message saying 'wait for host to start'

**--------------------------------------------------** Sends "Await simulation start"**----------------------------------------------->**

Message from player ignored until all players ready

'Start simulation' button enabled

click 'Start simulation'

Host panel & string grid removed

Start session

**<------------------------------------------------------** Marker = 4+username sent at 1 sec intervals-------- **--------------------------------------**

CouplingFileLoadedAwaitingPlayersFlag = false

HostInSessionFlag = true

PlayerAwaitingHostStartFlag = false

PlayerInSessionFlag = true

Join panel removed

Start session

**---------------------------------------------------** Sends "Start session" **--------------------------------------------------------->**

2

**----------**-Sends TTClockTime+DynMapFromHost as buffer--------**>**

Updates DynMapToHost every 5 secs

**<--------**-Marker = 5+username+DynMapToHost as buffer---------

Updates HostCombinedDynamicMap, updates grid & InfoVector

Updates TTClock + entry times & services. If 30 secs with no contact self-updates times to entry, if 5 mins & no contact player drops out.

If the self-update involves a train entering, it enters, and every time

a train enters normally it is checked to see whether it has already

entered and if so it is ignored.

If there are no changes to report the request is ignored

If there are then a change message is sent

Asks if there are any timer changes (every second)

**<-----** -------- --------**---**-Marker = 6+username -------- -------- ---------

**----------**-Sends appropriate change message --------**>**

Acts on the change (ignores if it has already been acted on)

**Player cancels join request**

End sequence, player can attempt to rejoin

PlayerCancelJoinFlag = false

Player clicks 'Cancel' button on panel (panel only visible before session begins)

Remove player from grid & InfoVector

**<----------**-Marker = 3+username sent at 1 sec intervals ------ -----

PlayerCancelJoinFlag = true

All other player flags = false

**-----------------------------**-Sends "Cancelled"----- -----**------**------------**>**

Actions during session not listed above:

Host actions other than as a participant:

Pause, change speed, save session (for all players)

Masterclock not disabled when deactivated

Player actions:

Acts on host commands to pause, change speed or save session

Masterclock not disabled when deactivated

Times to entry from DynMapFromHost used in place of timetable times for floating window