Communication Protocol

Client and server communicate via JSON messages.

The use of JSON messages allow to build clients using whatever

Each client request/action sent to the server is followed by a response to each client if it is a broadcast message (e.g. event on the map) or to the original sender (e.g. “legal/illegal action”).

The server sends messages to request setup data and possible action like move or build.

Example Message:

Client moving builder from (0,0) to (0,1)	Server notifies all clients (if there was no problem):

{
“type”: “move”,
“src”: {
“i”: 0,
“j”: 0
},
“dst”: {
“i”: 0,
“j”: 1
}
}
	{
	“type”: “move”,
	“name”: “player1”,
	“src”: {
	“i”: 0,
	“j”: 0
	},
	“dst”: {
	“i”: 0,
	“j”: 1
	},
	“result”: true
	}

After every move or build event the server sends a message containing the list of every suitable Cell for moving or building during next step. In this way the client knows which option is available. After each step the model looks at the player next state, if it is both move and build it asks to choose between the two. Last option is that the server notices that the player could do both move and play. In this last case the server asks to choice between the two option.

These requests are optional for the client, meaning that it can ignore them and ask for whatever it wants (both move and build and with every desidered cell). If the action is accepted the server notifies all client map changes, otherwise only the calling client is notified that the action has been refused.

There are two main different section: setup and game.
Setup phase asks for: number of players, name, date, color, god card and where place builder at first for each player.

After the setup phase there is the game phase, where players send to the server actions like move or build until the server replies with an end game, which means that there is a winner.

In the next pages is shown an example of messages sent during an entire game

Setup number of players

Server → Client

AskNumberOfPlayers

{"type":"askNumberOfPlayers"}

Client → Server

SetNumberOfPlayers

{"numberOfPlayers":2,"type":"setNumberOfPlayers"}

Setup nickname and date

Server → Client

AskNickname&Date (broadcast)

{"type":"askNickAndDate"}

Client → Server

Nickname&Date request

{"date":"1998.04.11","name":"pitty","type":"addPlayer"}

Server → Client

Nickname&Date reply

{"result":true,"name":"pitty","type":"playerAdded"}

Setup Color

Server → Client

AskColor request

{"chosenColors":[],"type":"askColor"}

Client → Server

ColorUpdate request

{"color":"MAGENTA","type":"colorUpdate"}

Server → Client

ColorUpdate reply

{"result":true,"color":"MAGENTA","name":"pitty","type":"colorUpdate"}

Setup Cards

Server → Client

AskGod request

{"godDescriptions":{"Atlas":"Your Worker may build a dome at any level","Zeus":"Your worker may build a block under itself","Apollo":"Your Worker may move into an opponent Worker\u2019s space by forcing their yours just vacated. ", ...},"chosenGodCards":[],"type":"askGod"}

Client → Server

SetGodCard

{"godCard":"Minotaur","type":"setGodCard"}

Server → Client

Server Reply

{"godCard":"Minotaur","result":true,"name":"pitty","type":"godCardAssigned"}

Setup Builders

Server → Client

AskBuilders

{"type":"askBuilders"}

Client → Server

BuildersPlacement

{"name":"pitty","positions":[{"i":1,"j":1}, {"i":2,"j":2}],"type":"buildersPlacement"}

Server → Client

BuildersPlacement reply

{"result":true,"name":"pitty","positions":[{"i":1,"j":1}, {"i":2,"j":2}],"type":"buildersPlacement"}

Move

Server → Client

Possible Move destinations

{"possibleDst":[[{"i":0,"j":1},{"i":1,"j":0}],[{"i":3,"j":3},{"i":3,"j":4},{"i":4,"j":3}]],"type":"possibleMoveDestinations"}

Client → Server

Move

{"dst":{"i":0,"j":1},"src":{"i":0,"j":0},"name":"pitty","type":"move"}

Server → Client

{"result":true,"dst":{"i":0,"j":1},"src":{"i":0,"j":0},"name":"pitty","type":"move"}

Build

Server → Client

Possible Build destinations

{"possibleDst":[[{"i":0,"j":0},{"i":1,"j":2},{"i":0,"j":2},{"i":1,"j":0}],[{"i":3,"j":3},{"i":3,"j":4},{"i":4,"j":3}],[],[]],"type":"possibleBuildDestinations"}

Client → Server

{"dst":{"i":1,"j":2},"src":{"i":0,"j":1},"buildDome":false,"name":"pitty","type":"build"}

Server → Client

{"result":true,"dst":{"i":1,"j":2},"src":{"i":0,"j":1},"buildDome":false,"name":"pitty","type":"build"}

Ask Step

Server → Client

AskStep

{"type":"askStep"}

Client → Server

SetStepChoice

{“stepChoice”: “MOVE”, “type”:”setStepChoice”}

Server → Client

Possible Move destinations

{"possibleDst":[[{"i":0,"j":1},{"i":1,"j":0}],[{"i":3,"j":3},{"i":3,"j":4}, {"i":4,"j":3}]],"type":"possibleMoveDestinations"}

Loss

Server → Client

LossUpdate

{“type”:”lossUpdate”)

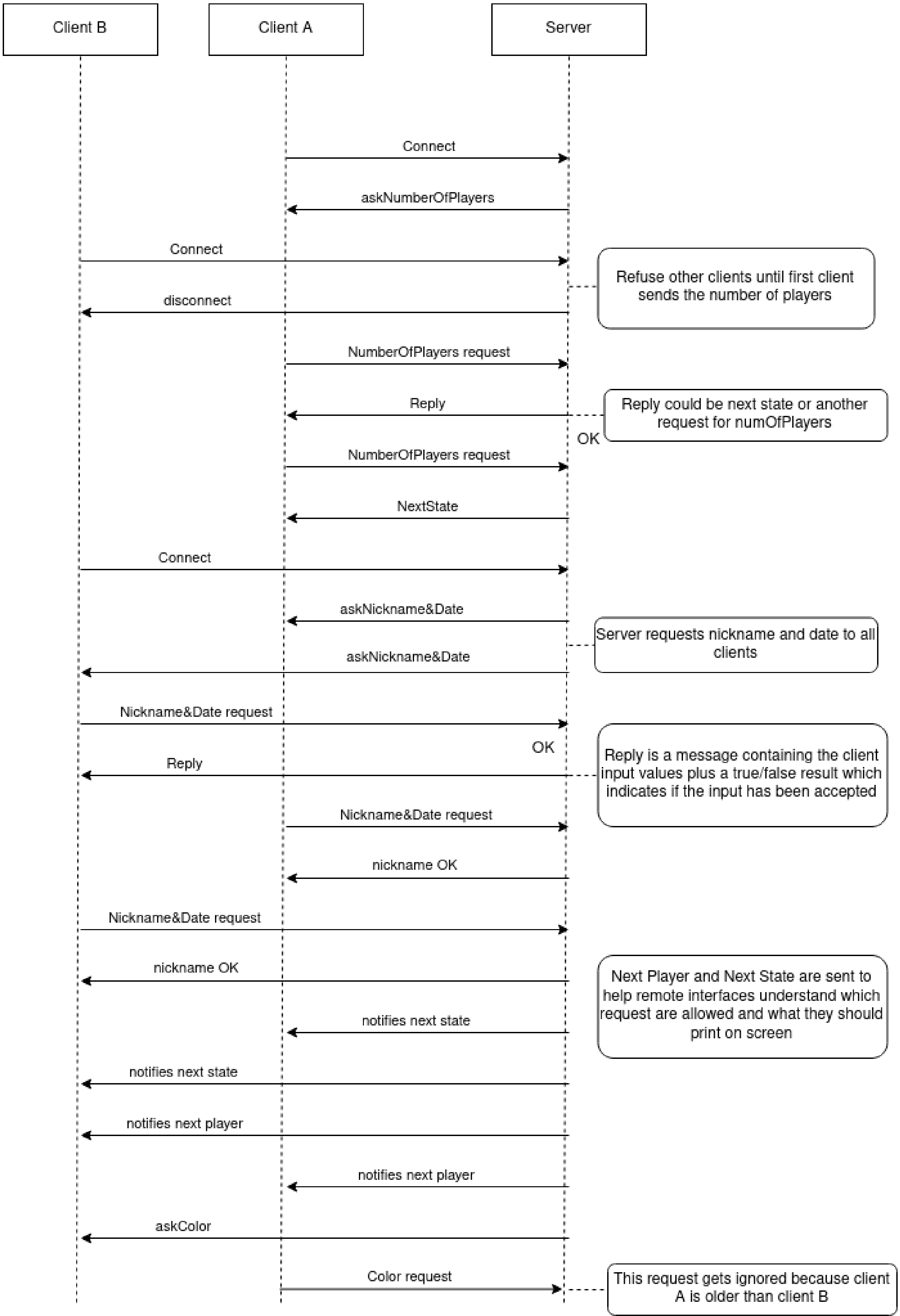
Game Over

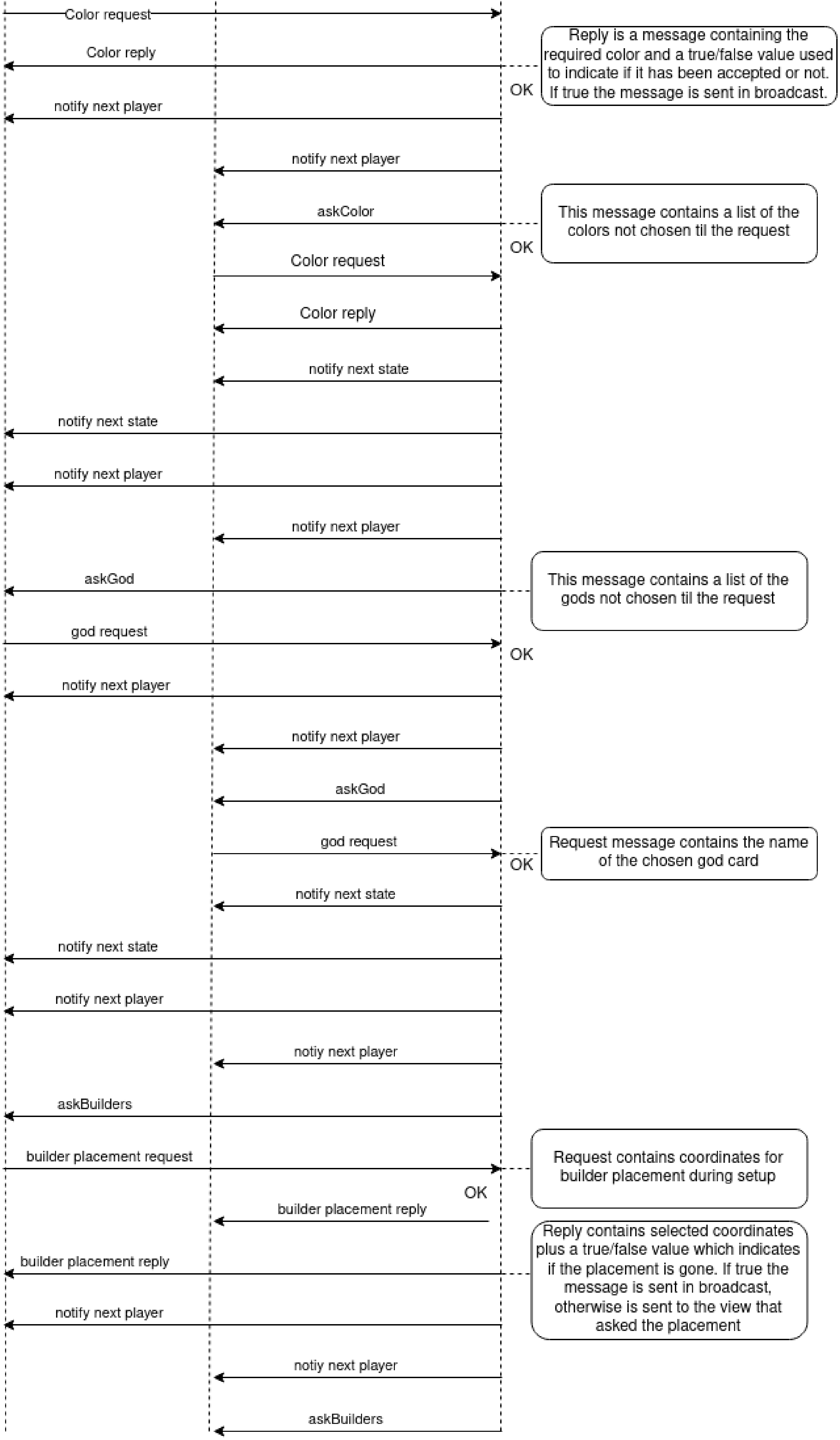
Server → Client

EndGame

{“type”:”endGame”,”winner”:”pitty”}

We suppose that client B is younger than client A, so the turn will start by A


Client B
Client A
Server
Connect
askNumberOfPlayers
Connect
Refuse other clients until first client sends the number of players
disconnect
NumberOfPlayers request
Reply
Reply could be next state or another request for numOfPlayers
OK
NumberOfPlayers request
NextState
Connect
askNickname&Date
Server requests nickname and date to all clients
askNickname&Date
Nickname&Date request
OK
Reply
Reply is a message containing the client input values plus a true/false result which indicates if the input has been accepted
Nickname&Date request
nickname OK
Nickname&Date request
nickname OK
Next Player and Next State are sent to help remote interfaces understand which request are allowed and what they should print on screen
notifies next state
notifies next state
notifies next player
notifies next player
askColor
Color request
This request gets ignored because client A is older than client B



Reply is a message containing the required color and a true/false value used to indicate if it has been accepted or not. If true the message is sent in broadcast.

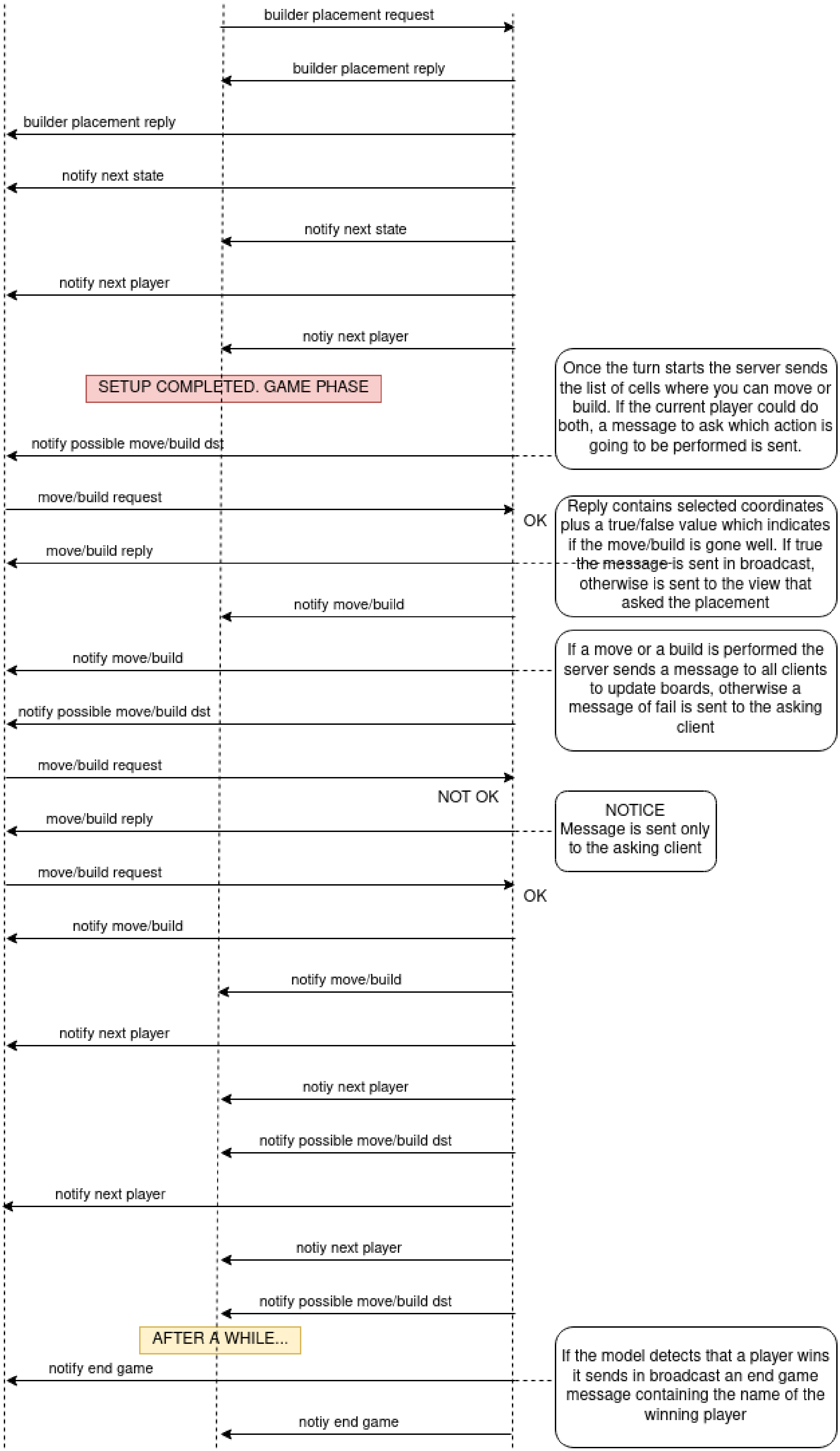
This message contains a list of the colors not chosen til the request

This message contains a list of the gods not chosen til the request

Request message contains the name of the chosen god card

Request contains coordinates for builder placement during setup

Reply contains selected coordinates plus a true/false value which indicates if the placement is gone. If true the message is sent in broadcast, otherwise is sent to the view that asked the placement



Once the turn starts the server sends the list of cells where you can move or build. If the current player could do both, a message to ask which action is going to be performed is sent.

Reply contains selected coordinates plus a true/false value which indicates if the move/build is gone well. If true the message is sent in broadcast, otherwise is sent to the view that asked the placement

If a move or a build is performed the server sends a message to all clients to update boards, otherwise a message of fail is sent to the asking client

NOTICE
Message is sent only to the asking client

If the model detects that a player wins it sends in broadcast an end game message containing the name of the winning player