Over<https://portal2.gamesparks.net/games/q370455D15Az/config/cloudcode>Reaction Network Schema

Summary:

The OverReaction game will need to pass data between the players to maintain a consistent simulation model. When playing a match against another player, one player will become the “Host”. The host will update the server as needed with the status of the current simulation. The Opponent will then be updated of the changes. The opponent will not have to run the simulation.

Glossary:

* Host: The Player who is running the simulation locally. (Used in the context of an individual match)
* NHPlayer: NotHostPlayer The Player who does NOT run the simulation locally. (Used in the context of an individual match)
* Player: A user who will play the game.
* Client: Any players machine, host or Opponent
* Server: The “cloud” that will handle login, matchmaking and passing data between the Host and the Opponent.

Network Realtime Opcodes:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OpCode | Name | Direction | Desc | Return opcode |
| 1 | PlayerJoin | Server->  Client | When a player joins, the server sends a packet with code 1. This contains the displayName of the player who joined. | N/A |
| 10 | GetHostID | Client -> Server | Asks the server the ID of the host | 11 |
| 11 | HostIDResponse | Server ->  Client | The response sent to a client when they send a GetHostID packet.  Returns the host ID. | N/A |
| 20 | PlayCardRequest | Client->  Server | Sends a message to the server asking to play a card. If the player can play a card and is the NHPlayer, then a code 24 is sent to the host. | 21  ?24 |
| 21 | PlayCardResponse | Server->  Client | Response after trying to play a card. Contains a boolean of whether or not the card successfully played and a String message for a reason. | N/A |
| 22 | ReactionApplied  (OBSOLETE) | Host -> Server | Indicates the card that has been added to the simulation | 23 |
| 23 | ReactionAppliedAlert (OBSOLETE) | Server->  NHPlayer | Indicates to a non host player that a card has been added to the simulation. | N/A |
| 24 | PlayCardRequest? | Server->  Host | Notifies the host that a NHPlayer wants to play a card. | N/A |
| 25 | SendInitialDecks | Host->  Server | Sends the generated decks after initialization to the server to be tracked and sent to opponents | N/A  26 |
| 26 | InitialDeck | Server ->  NHPlayer |  |  |
| 100 | UpdateGameState | Host->  Server | This is used to send the game state (Contents, Reactions) to the NHPlayer. | N/A |
| 101 | SendGameState | Server->  NHPlayer | This packet is used to relay the game state to the NHPlayer in order to keep the game synced. | N/A |

How Gamesparks Real Time packets are laid out.

OpCode: Each packet has an OpCode. This opcode allows us to specify the purpose of a packet.

Data: Each packet contains an array that can take any type of data. Although since it is one array that can store many different types of data we need to keep track of what type is where. Otherwise we risk throwing a bunch of errors. The way that the data is managed is with the use of helper methods like setDouble(int index, double value) and getDouble(int index).

NOTE: The problem come from, for example, using getDouble on an index containing a string. To try to mitigate this we will document the layout of the data for each packet.

Packet Layouts

11: GetHostID Response

Data:

|  |
| --- |
| 1 |
| String: PeerID of the host |

20: PlayCardRequest

Data:

|  |
| --- |
| 1 |
| Int: position of card in the players deck |

21: PlayCardResponse

Data:

|  |  |
| --- | --- |
| 1 | 2 |
| Int: boolean representing if you can play the card. 0=false, 1=true | Int: Position of the card in the players deck |

24: HostPlayCard

Data:

|  |  |
| --- | --- |
| 1 | 2 |
| Int: Position of the card in the players deck | The peerID of the player playing the card. (used for the host to get the players deck) |

25: SendInitialDecks (this packet goes to the server before being delegated into one 26 for each player)

Data:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Repeat For Each Player | | | |
| Once Per player | | For Each Card in players deck | |
| 1 | 2 | 3 | 4 | 5 |
| Int: num players | Int: Length of the deck | String: PeerID of the player | String: Card Display String | Int: Position in deck |

26: SendInitialDeckToPlayer

Data

|  |  |  |
| --- | --- | --- |
|  | Repeated for each card | |
| 1 | 2 | 3 |
| Int: length of deck | String: Card Display String | Int: position in deck. |

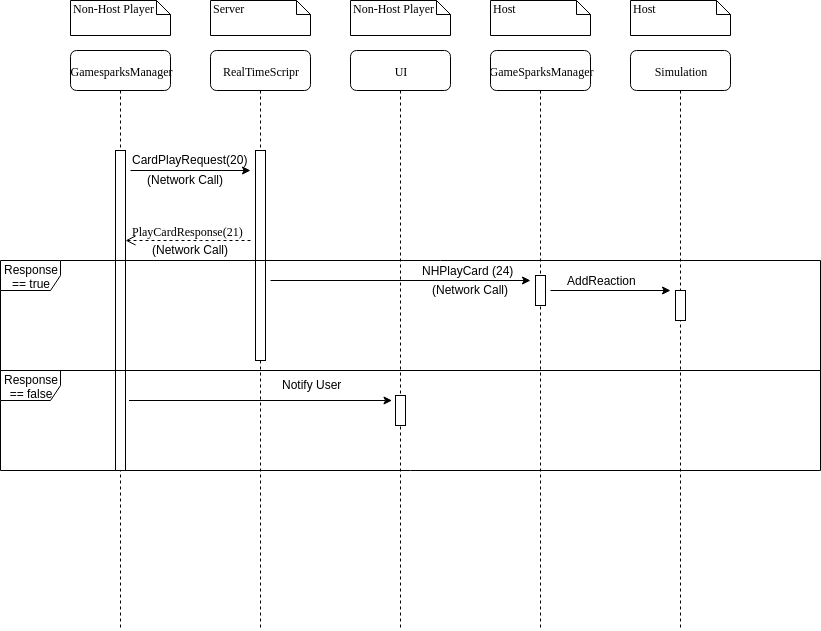
100: UpdateGameState

Data:

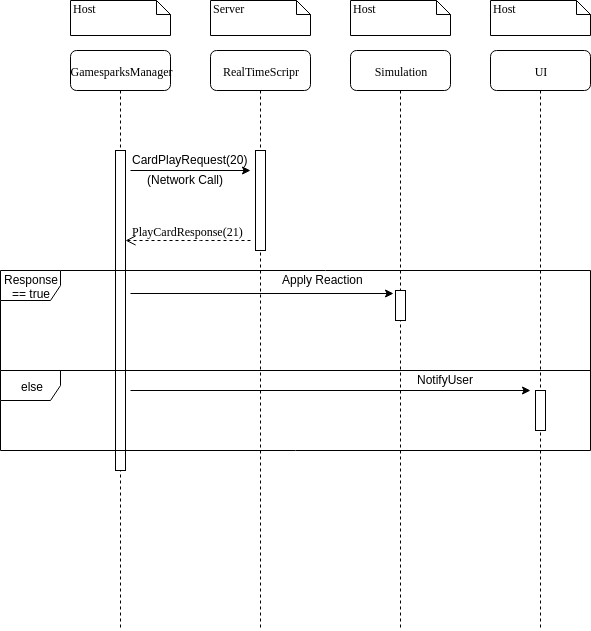
|  |  |  |
| --- | --- | --- |
| For Each Live Species | For each live reaction in the simulation | |
| 1 .. NumLiveSpecies + 1 | numLiveSpecies + 2... | numLiveSpecies + 3 |
| Int: the quantities of a each live species | String: the equation of the reaction to display | Double: the remaining time left for the reaction |

# Sequence Diagrams

PlayCardRequest (Non-Host Player)



PlayCardRequest (HOST)



Send Game State

