Swift/Act-r communication through action module:

* ***Situation: The first round for the player.***
  + Input:
    - State: **first\_start**
  + Output:
    - Card\_ask
    - ~~Card\_get~~
    - Current\_player
    - Opponent\_player
    - state\_round: state of the round (succeed/gofish)
    - first\_starter (default false)
    - **State: state of self (wait\_memorize /gofish/succeed/start)**
  + Action outside production rules:
    - One random player will be selected to take the first act (ask the player to his left for the first card at hand). Change the first\_starter to true for the first player.
    - The first act will happen automatically when receive “first\_start” state. Then the state of the player will be changed to “gofish” or “succeed”. The state of other players will be changed to “wait\_memorize”
    - When the round state is “**gofish**”, swift logic needs to decide who is the next to player and change players’ states accordingly
* ***Situation: Not your round.*** 
  + Input:
    - State: **wait\_memorize**
  + Output:
    - Card\_ask
    - ~~Card\_get~~
    - Current\_player
    - Opponent\_player
    - state\_round: state of the round (succeed/gofish)
    - **State: wait\_memorize (next round is still wait), start (you will initiate the next round --- go to the strategy production rules)**
  + Action outside production rules:
    - When the round state is “**gofish**”, swift logic needs to decide who is the next to player and change players’ states accordingly.
  + Actions in act-r:
    - Creating memory chunks for player/card relationship
* ***Situation: ask a player for a card.*** 
  + Input:
    - State: **ask**
    - Card\_ask
    - Opponent\_player
  + Output:
    - Card\_ask
    - ~~Card\_get~~
    - ~~Current\_player: self~~
    - Opponent\_player
    - Set (whether the asking card is a set or not)
    - State: state of self (gofish/succeed)
  + Actions in act-r:
    - Creating memory chunks for exposed cards
* ***Situation: check card at hand.*** 
  + Input:
    - State: **checking**
    - card\_deck: first (ask for first card) or =card (ask for the card after =card) or multiple (check for multiple cards)
    - ~~strategy: identifier for the current strategy (i.e., g1, g2, aggressive, conservative)~~
  + Output:
    - card\_deck (in the “multiple” case, if no multiple cards, send back the first card)
    - ~~strategy~~
    - State: checking or checking\_failed (reached the last card)
* Other interactions:
  + For each player, map the opponents’ name to player 1, player2 and player3 for usage in production rules.