DATING GAME - http://cs.nyu.edu/courses/Fall12/CSCI-GA.2965-001/dating.html

---

Architects : rky
Randy - rjs471@nyu.edu, Kunal - kunal.balani@nyu.edu, Yuriy - yuriy.skobov@nyu.edu

GAME MECHANIC

---

1) P submits a vector of weights.
2) The server generates 20 random candidates scored using the weights provided by P.
3) M receives the 20 random candidates.
4) M iteratively submits up to 20 candidates.
5) At each submission, P submits 'noise' for weights.
6) Each candidate is scored against the altered weights.
7) Either the M guesses the ideal candidate in < 20 attempts, or the game concludes.
8) M gets a score.

MESSAGE PROTOCOL

S = Server, M = Matchmaker, P = Person

```
GAME INIT
  P->S:  team1                                                    # Team name
  M->S:  team2                                                    # Team name
  S->M:  M 7                                                      # Player, |Attributes|
  S->P:  P 7                                                      # Player, |Attributes|
  S->P:  WEIGHTS                                                  # Timer on P starts
  P->S:  [-0.20, -0.45, -0.35, 0.28, 0.22, 0.0, 0.5]              # P's weights
  S->P:  OK | ERROR "error message"                               # Timer on P pauses
  S->M:  [0.0, 0.0, 0.1, 0.0, 0.50, 1.0, 1.0] 0.57, [0.0, ...] 0.23  # 20 scored, random candidates

GAME LOOP (<=20x)
  S->M:  CANDIDATE                                               # Timer on M starts
  M->S:  [0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0]                     # Proposed candidate
  S->M:  OK | ERROR "error message"                              # Timer on M pauses
  S->P:  NOISE                                                   # Timer on P starts
  P->S:  [0.01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]                    # Weight 'noise'
  S->P:  OK | ERROR "error message"                              # Timer on P pauses
  S->M:  -0.09                                                   # Score using altered weights

GAME RESULT
  S->P:  GAMEOVER -0.09 20                                       # Score using unaltered weights
  S->M:  GAMEOVER -0.09 20                                       # Best score, number of guesses
```

ADDITIONAL NOTES

- Noise is not cumulative; it is applied against the weights supplied by P in step 1.

- Noise is a vector of weight displacements. |noise| == |weights|

- When generating noise, P may modify 5% of the criteria (P may choose which ones)
  by 20% each, e.g. if a chosen criterion has a weight of 0.4, then P can modify it
  to any value between 0.4 - (0.2*0.4) and 0.4 + (0.2*0.4). This must be within [-1, 1] range

- P supplies noise without knowing anything about the candidates proposed by M.

- Score for a candidate is the dot product of the weights and the candidate vectors.

- A player X beats player Y if X obtains a greater score or obtains the same score but
  with fewer candidates than Y does for X's P.

- This game is turn-based, don't calculate while not your turn.

- All communication will be via a socket. Your program should take a port number as an argument

- 'P' can't 'win'. The game result is a measurement of how well M performs.
  As a result, if P submits invalid data, then M advances, but if M submits invalid data
  then he gets a score of -1.0, 20. Moreover, client program errors are game terminating.

- In any case where this document differs from the web page, this document is authoritative.