

# Competitive Behavioral assessment of Rivalry and Admiration (CoBRA) task – V2 Manual

---

## Contents

Short description.....	1
Installation .....	2
Instructions for task administration .....	2
Modifying the code: .....	5
Troubleshooting:.....	5
General codebook.....	6

## Short description

The CoBRA task is a social decision-making task designed to behaviorally capture narcissistic reactions to defeat, namely narcissistic rivalry and admiration-seeking, in a computer game competition where the outcomes are rigged. Players are told they will play against previous participants whose performance has been recorded and that in doing so, they will take part in the competition's global ranking. For a detailed description of the task, see the following article and its supplement (for any questions regarding the task, please contact the corresponding author, [Anna Szücs](#)):

- ➔ **Szücs A**, Wright AG, Edershile EA, Dombrovski AY, Rivalry and admiration-seeking in a social competition: from traits to behaviors through contextual cues. **Accepted by *Personality Disorders: Theory, Research, and Treatment***.

Competitive behaviors are measured by 2 distinct outcome measures:

- outcome measure 1 – apple/point stealing – behavioral rivalry: on each trial, before competing, participants can choose to steal points from their assigned opponent's score on that trial.
- outcome measure 2 – rank buying – behavioral admiration-seeking: at the end of each round, participants can pay for a booster that improves their rank in the competition's global ranking.

In this second version (V2.6), the task's design allows to play up to four blocks of 36 trial each with different defeat:victory ratios, which define *extreme losing* (3 defeats to 1 victory), *moderate losing* (2 defeats to 1 victory), *winning* (1 defeat to 2 victories) or *neutral* (1 defeat to 1 victory) conditions. You can choose to play only some of these blocks, or all of them. If you choose to play less than 4 blocks, the task will select the first ones per the enumeration above by default. You can change these to different ones in the code (see the Modifying the code section of this manual)

*Please, note that this game is designed to work on PCs and Windows tablets, the code might need adaptations to run on a Mac.*

## Installation

- Decompress the .zip file called “Snake game V2-6-main.zip” in a location of your choice.  
**Warning:** Do not put its content in MATLAB’s folder; anywhere else is fine.
- Make sure you have MATLAB installed on your computer (version 2013 or older) as well as Python version 2.7.

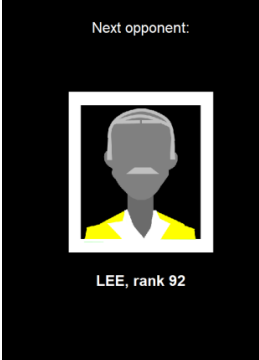
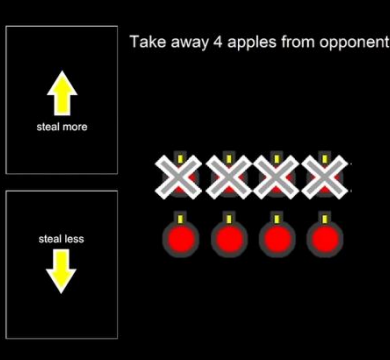
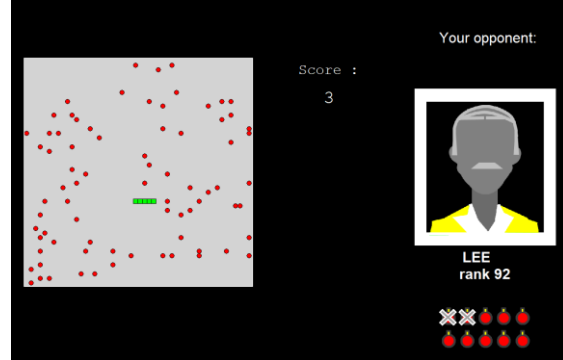
## Instructions for task administration

- Each block is about 12 minutes long, so make sure your device is plugged in, or that it has enough battery.
- Sound must be turned on; check that the volume is audible without being too loud.
- Open the “cobra\_task\_V2-6” folder. Once you are in the folder, open the file 'main\_game\_launcher.m' with MATLAB (right click -> Open with -> MATLAB), and simply click “run” from the MATLAB editor window (green arrow in the upper menu).
- First, go through “Competition Rules” with the participant. It is really important not to say anything that would destroy the illusion of a real competition. Ask the patient whether they have any questions at the end of the rules. Do not answer anything about what strategy they should use; just make sure that they understood all the rules correctly. If it is not the case, feel free to cycle through the rules one more time.
- Second, go to “Snake practice”. Emphasize to participants that there are only 2 command keys in this game: the left and the right arrow key. The snake turns to its own left each time they press left (it is not necessarily their own left depending on how the snake is oriented on the screen). Same thing with turning right.  
If the participant wants to, they can practice a second time before entering the competition.
- You can then start the game by pressing “START”:
  - Select “36” for the number of rounds (this corresponds to the total number of trials per block; the two lower numbers are only there for troubleshooting purposes).
  - The ID number should be the patient’s ID in the lab, but any other string of characters will work (do not use special characters though).
  - Participants can enter any name (middle names, nicknames or anything else they feel comfortable with). To increase the game’s credibility, you can ask them to enter a pseudonym that would preserve their confidentiality towards the other players who will play after them (in reality, their player names will not be added to the ranking after they have played).
  - If the task does not let you go to the next page, it is because some data is missing.

**Warning:** The folder with the .csv files containing the participant's data is created based on this ID page (the file's name is the participant's ID number). So, do not enter the same ID several for different participants, else previous files will be overwritten. This makes it also really important to backup all collected data elsewhere before starting a new experiment!

- On the next page, participants will have to choose their avatar for the game by pressing the little rectangle with the corresponding number under the chosen picture. They have to choose whether they agree to being listed in the ranking after they played (this is also part of the deception to increase credibility; the ranking and opponents are the same for all participants).
- Next, participants will have to fill out some items of the PANAS affective scale (incorporated in the game) and a few other questions (see Codebook below). They will have to do it at the beginning and at the end of the game. If they miss out one or two answers (by clicking too quickly on the screen), just carry on with the experiment, this item will be automatically coded as "0" in the .csv file (this can be later recoded as NA as valid ratings go from 1 to 5).
- The competition starts: participants will see their next opponent for a few seconds (Screen 1 below). Then, they can choose to take away apples from their opponent (Screen 2 below). Next, the snake game window appears and they can start the snake whenever they are ready by pressing the left arrow key (Screen 3 below).

**Warning:** if participants inadvertently click on the screen instead of clicking on the left arrow key, the window containing the snake may disappear. If this is the case, see the second bullet point in the troubleshooting section of this manual.

		
<p><b>Screen 1:</b> <u>Next opponent</u> (goes automatically to 2nd screen after a few seconds)</p>	<p><b>Screen 2: Apple/point stealing</b> (pressing the arrows increases or decreases the number of points stolen; the default is the middle option, 4 points/apples)</p>	<p><b>Screen 3: Playing snake</b> (the apples below the opponent's picture are a reminder of the player's choice for apple stealing on that same round)</p>

- After the snake round is over, a screen will be displayed with the results, paired with a winning or losing sound (Screen 4 below). When participants are ready, they can move to the next screen where they can choose to purchase rank-enhancing boosters if they want to (Screen 5 below). The final window shows their rank with the standard rank gain/loss + the chosen booster's effect

(Screen 6 below). Their position at the end of the round is highlighted in yellow in the ranking. Then, they automatically move to the next round (i.e. trial).

**Warning:** Watch the participants' one or two first rounds, and then let them play on their own. It is very different how people make decisions when they think that nobody is looking.

<p><b>Screen 4:</b> <u>Outcome window</u> (the bars in the middle only indicate the final apple count; participants do not know how many apples were stolen from them)</p>	<p><b>Screen 5: Standard rank gain and possibility to by boosters to increase one's rank.</b> (pressing the arrows increases or decreases the number of rank boosters bought; the default is the middle option, the Silver booster granting 2 extra ranks)</p>	<p><b>Screen 6:</b> <u>Additional rank gain with a booster</u></p>

- At the end of each block, participants will have to fill out some items of the PANAS affective scale again as well as a few additional items (see Codebook below). Then, after seeing a screen with their final rank and monetary earning, they will have to answer a series of end-game questions that are also incorporated in the game.
- Pay the participants \$5 as a real monetary reward (or any other amount that was decided in advance; see the Modifying the code section for how the amount can be changed), without letting them know that this is a fixed amount. For the sake of credibility of potential subsequent experiments, it is important to keep the illusion that it has been computed based on what the participant has saved on three random trials.  
**Warning:** If possible, do not disclose any information about the game's hidden design to participants before the whole cohort has played it.
- Centralize the .csv files generated during the game by copying them to a safe location after each participant.  
**Warning:** the .csv files are semicolon delimited; they might look a bit messy in Excel when opened on some tablets/computers but this is just a matter of Excel settings: select the first column of your data in Excel, then go to "Data > Text to columns" and change the delimiter to semicolon.

## Modifying the code:

- The code is structured in the following way:
  - **main\_game\_launcher.m**: this script is where you can modify most settings. It is thoroughly annotated to make this easier.
  - **functions folder**: this folder contains all other scripts used by the task, in .m and .py formats, as the 'snake game' portion of the game is coded in Python.
  - **medias folder**: this folder contains all pictures used in the task
  - **data folder**: this folder is created upon playing the task for the first time and contains all data collected from participants. Each participant has a subfolder named with their ID number. Inside, you will find a .csv file of all collected variables (see codebook below) and individual text files with time stamps in seconds for right arrow key presses (coded as 1), left arrow key presses (coded as 2) and apples eaten (coded as 8).
- **Changing participants' monetary gain**: if your participants will receive a different amount than \$5, open the '**main\_game\_launcher.m**' script. Under % money gain, change the final\_earning variable from '5' to your amount. Save your changes. Please note that depending on the amount you will choose, you may also want to change participants' virtual money endowment to make it proportional. You can do so just one line above (variable 'endowment').
- **Changing the types of blocks**: If you choose to play less than 4 blocks but different block types than the default, you can open the script "functions/**round counter.m**" and modify the variable block\_order following the instructions in the script's comments.

## Troubleshooting:

- If the code does not start in MATLAB, or if the "Snake practice" does not work properly, verify that MATLAB is in the right folder (in the folder 'Snake Game V6.2' where the file 'main\_game\_launcher.m' is). If it is not the case, go to **MATLAB's command window** and enter "cd" followed by a space, and then the folder's location.  
E.g.: " **cd** C:\Users\username\Desktop\cobra\_task\_v1-6) "
- Only the keyboard should be used during the snake game portions of the program (these are coded in Python and are not fully incorporated into the interface). Make sure to remind the participants that they are not supposed to put their finger on the screen/mouse except when pressing buttons (some people try to click more than necessary...).  
Most importantly, one must **\*not\*** click anywhere else on the screen while the window with the snake is visible. Doing so hides the snake game behind the other MATLAB windows. If this happens, the game will continue in the background, **so do not close any windows!** Just minimize all MATLAB windows, click anywhere on the snake game window and let the patient play the game normally. Once finished (after the usual 15 seconds timeout), you can restore all MATLAB windows to full size. The game is then back on track.
- If the game crashes, you will need to press "ctrl + C" in **MATLAB's command window**, in order to end the script completely. Once you did this, make a screenshot of the error message (or copy the error message somewhere else). Make a copy of the data collected until then. **Do not relaunch the game if the participant has already passed more than the first two rounds** because then they would know that the outcomes are rigged.

# General codebook

All variables are numeric, except for the ones specified as *categorical*.

## Task-level variables answered at the beginning of the task

**ID:** categorical variable; participants' ID (entered at the beginning of the task and also constituting the name of their personal folder with all their data files).

**n:** participant count (n = 1 for the first participant, n = 2 for the second participant, etc.)

**blocks:** number of blocks played (options are 1, 2 or 3).

**blockOrder:** categorical variable; the order of blocks played; coded as a three-digit combination of 1 = defeat++ block (3 defeats:1victory); 2 = defeat+ block (2 defeats:1 victory); 3 = winning block (1 defeat:2 victory); 4 = neutral block (1 defeat: 1 victory). E.g., a block order of defeat++ block → winning block → defeat+ block would be coded "132".

**name:** the pseudonym entered by the participant at the beginning of the task.

**gender:** categorical variable, with 1 = male, 2 = female, 3 = other

**age:** participants' age in years.

**gameExp:** Video game experience rated from 1 to 5 (participants are asked to count anything they played on tablets or smartphones, not only on computers): 1 = 'played < 5 times/lifetime'; 2 = 'played > 5 times/lifetime but last time > 10 years ago'; 3 = 'plays 1-5 times/year'; 4 = 'plays > 5 times/year but not weekly'; 5 = 'plays weekly'.

**avatarChoice:** categorical variable; participants' choice of 1 out of 4 avatars they are presented with. Depending of the gender they indicated. See **Table 5** for the avatar choices.

**consentChoice:** categorical variable. 1 = agrees to be listed in the ranking for subsequent players, 0 = does not agree.

**snakeLevel:** snake's speed in the arcade game established based on performance during the practice phase (20 seconds = same duration as all trials). If the practice phase has not been played, snakeLevel defaults to 2.

1 = easy/slow; if participant caught < 6 apples during the practice phase on their last attempt.

2 = medium; if participant caught 6-12 apples during the practice phase on their last attempt.

3 = difficult/fast; if participants caught > 12 apples during the practice phase on their last attempt.

**trial:** current trial on a given block (ranges from 1 to 36).

**trial\_tot:** total trial out of the entire game (ranges from 1 to 108).

## Block-level, randomized variables

**block\_num:** current block (1 = first block, 2 = second block, 3 = third block).

**blockCurrent:** categorical variable; the current block participants are playing in: "3d1v" = losing++ block with 3:1 defeat ratio; "2d1v" = losing+ block with 2:1 defeat-victory ratio; "1d1v" = losing+ block with 1:1 defeat-victory ratio; "1d2v" = winning block with 1:2 defeat-victory ratio.

### **Trial-level, subject-dependent variables**

**appleChoice:** *point stealing*, i.e., player's choice of the number of points they wish to steal from their opponent before playing. 1 = none, 2 = 1 apple, 3 = 2 apples, 4 = 5 apples, 5 = 10 apples. The player can move up and down between the options. The option selected as default is 3.

**rankChoice:** rank buying, i.e., player's choice of buying a booster to increase rank at the end of a given trial. 1 = none, 2 = + 1 rank, 3 = + 2 ranks, 4 = + 3 ranks, 5 = + 4 ranks. The player can move up and down between the options. The option selected as default is 3.

**score:** task performance (i.e., number of apples eaten by the snake) on the snake arcade game on a given trial. It's modulated by the participant's level (since their performance during practice influences the snake's speed), their agility and whether it is a winning or a losing trial (more apples are available on winning trials: 80 vs. 60 apples)

**rankStart:** player's rank at the end of the trial but before rank buying (participants start each block at rank 250 out of 500; they move up 4 ranks after each victory and move down 4 ranks after each defeat). Rank is coded as 500 = lowest rank, 1 = highest rank.

**rankEnd:** player's rank at the end of the trial but after rank buying. Rank is coded as 500 = lowest, 1 = highest rank.

### **Trial-level, subject-independent variables**

*Predefined design variables identical for all participants on a given block; see **Tables 1 to 4** below for the contingencies of predefined variables.*

**win:** categorical variable; a given trial's outcome: 1 = victory, 0 = defeat.

**close:** categorical variable; based on scoreDiff : 1 = tight competition (close-run; if scoreDiff  $\leq 2$ ; i.e., the absolute value of the score difference between the participant and their opponent is  $\leq 2$ ), 0 = large score difference between participant and opponent (displayed to players during the snake arcade game for the last 10 seconds of the game).

**scoreDiff:** absolute value of the score difference between participant and opponent (predefined variable).

**oppScore:** the opponent's score on a given trial, computed based on scoreDiff and win, such that oppScore = score +/- scoreDiff (only displayed to players in the form of a bar chart at the end of the trial).

**oppName:** categorical variable; the opponent's name on a given trial, displayed at trial-start.

**oppRank:** the opponents' ranks on a given trial, displayed at trial-start (see Tables 1 for all predefined variables). Opponent's rank is coded as 500 = lowest rank, 1 = highest rank.

**oppMale:** categorical variable; the opponent's supposed gender (based on name and avatar; this information is not displayed to players); 0 = female, 1 = male.

## Additional psychometric measures built in the task

### Block-level psychometric measures

**excited1 and 2, resentful1 and 2, ..., nervous1 and 2, misunderstood1 and 2:** participants are asked at the beginning of the task (e.g. excited1) and at the end of each block (e.g. excited2) to what extent they feel each affect; their answers are recorded as follows: 0 = “Very slightly/Not at all”, 1 = “A little”, 2 = “Moderately”, 3 = “Quite a bit”, 4 = “Extremely”;

7 vulnerable affects: insecure, resentful, underappreciated, fragile, vengeful, misunderstood, ignored

7 grandiose affects: glorious, powerful, prestigious, superior, extraordinary, brilliant, unrivaled

6 positive affects (PANAS): excited, proud, alert, determined, happy, relaxed

6 negative affects (PANAS): angry, sad, hostile, irritable, ashamed, nervous

**standingEstim1 and 2:** Participants’ estimation after the practice round (*rankEstim1*) of their final standing in the competition’s global ranking; and their estimation after each block of what should have been their standing (*rankEstim2*). Possible answers: 1 = “At the very bottom”, 2 = “In the lower tier”, 3 = “In the middle”, 4 = “In the higher tier”, 5 = “At the very top”.

**mot1 ... mot4:** motivation questions answered at the end of each block, assessed on a Likert scale: 1 = “Strongly disagree”, 2 = “Disagree”, 3 = “Neither agree nor disagree”, 4 = “Agree”, 5 = “Strongly Agree”;

mot1: 'I felt the need to maximize my own performance.'

mot2: 'It was important that I did better than others.'

mot3: 'I did not want to perform more poorly than others on the task.'

mot4: 'My main goal was to be on top.'

### Task-level psychometric measures

**oppEstim1 ... oppEstim4:** categorical variable; at the end of the task, participants are asked whether the 8<sup>th</sup> opponent of each block has defeated them (they see the opponent’s avatar and name as a reminder). 1 = “yes”, 0 = “no”, 9 = “maybe”, NaN = this block had not been played.

*oppEstim1:* 8<sup>th</sup> opponent of the defeat++ block (3 defeat: 1 victory ratio)

*oppEstim2:* 8<sup>th</sup> opponent of the defeat+ block (2 defeat: 1 victory ratio)

*oppEstim3:* 8<sup>th</sup> opponent of the neutral block (1 defeat: 1 victory ratio)

*oppEstim4:* 8<sup>th</sup> opponent of the victory block (1 defeat: 2 victory ratio)

**enjoyed:** at the end of the task, participants answer on an analog scale (slider button) from 1 to 10 to “How much did you enjoy playing?”

**satisfied:** at the end of the task, participants answer on an analog scale (slider button) from 1 to 10 to “How well do you think you played?”

**fair:** at the end of the task, participants answer on an analog scale (slider button) from 1 to 10 to “On average, how fair did your opponents play?”

**credible:** at the end of the task, participants answer on an analog scale (slider button) from 1 to 10 to “To what extent did your actions influence the outcome of the game?”



Table 1. Defeat++ block (3 defeats:1 victory ratio)						
oppMale	oppName	oppRank	scoreDiff	win	trial	
0	Shirley	403	18	0	1	
0	karen77	15	8	0	2	
1	LEE	444	9	0	3	
1	Ed	245	5	1	4	
1	Nate	289	2	1	5	
1	BRAD	92	17	0	6	
0	emma	120	2	0	7	
0	taylor	184	8	1	8	
1	anthonyY	490	1	0	9	
0	linda	357	7	1	10	
1	George	249	2	0	11	
1	matt	236	17	0	12	
0	Vivian	30	1	0	13	
0	mona	341	9	0	14	
1	ScoTT	22	5	1	15	
0	Carrie	36	6	0	16	
1	kev88	261	8	0	17	
0	Janice1	49	2	0	18	
0	rebecca	410	10	0	19	
1	STEVE	293	14	1	20	
0	Pam	362	5	0	21	
1	larry	75	16	0	22	
0	kim	330	1	1	23	
0	TESS-123	260	8	0	24	
0	L-I-S-A	487	0	0	25	
1	brand	325	4	0	26	
1	JoSH1	228	0	0	27	
1	HenRy	227	3	0	28	
1	BRiaN	117	2	1	29	
1	Vince	313	5	0	30	
0	TraceY4	42	3	0	31	
0	LLyynn	67	2	0	32	
0	K8Te	87	18	1	33	
1	adri4n	196	12	0	34	
0	Alexa	416	19	0	35	
1	Julian-0	402	1	0	36	


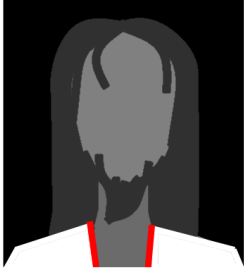

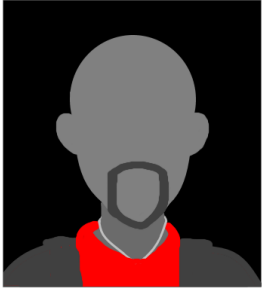




Table 2. Defeat+ block (2 defeats:1 victory ratio)					
oppMale	oppName	oppRank	scoreDiff	win	trial
1	dexter	261	1	1	1
0	emms20	270	18	0	2
0	MEGANE	250	2	0	3
0	Scarlett	62	7	0	4
1	MIKE	105	13	1	5
1	barryyy	53	15	0	6
0	elisa	283	0	1	7
1	Warren	62	9	0	8
1	AnDrEW	318	17	0	9
0	Faith	246	1	0	10
0	tina-1000	103	7	1	11
0	Valery	268	11	0	12
1	Pete99	223	2	0	13
0	JeSSy	437	13	1	14
1	HUGH	136	10	0	15
1	cyril-11	22	17	1	16
1	logan	74	2	0	17
0	rita	348	9	0	18
1	-IAN-	95	12	0	19
1	NiCO	427	3	1	20
0	Su-88	249	12	0	21
0	GAYLE	84	1	1	22
0	Patty	466	12	0	23
1	greg	365	17	0	24
0	lena	369	10	1	25
0	Gabriela8	32	9	0	26
1	Jase2939	431	2	0	27
1	Zach	103	0	1	28
0	Irls	209	15	0	29
1	d4rryl	474	16	1	30
0	elinor	42	2	0	31
0	Amber	321	1	0	32
1	lance44	72	17	0	33
0	sophiA	287	17	1	34
1	EVAN	311	2	0	35
1	maXim	27	13	0	36

Table 3. Winning block (1 defeat:2 victory ratio)

oppMale	oppName	oppRank	scoreDiff	win	trial
0	Deborah	364	9	1	1
1	Justin	481	1	1	2
1	mark	291	18	0	3
0	Vicks99	59	12	1	4
1	Alyson	29	9	1	5
1	marta99	343	0	0	6
1	GWEN	490	11	0	7
0	RoBB	298	14	1	8
0	peRRy	93	2	1	9
1	victoria	171	20	1	10
0	Edward	482	1	1	11
0	orlando	97	18	0	12
1	Jerome	467	11	1	13
0	THESSA	196	1	0	14
1	FeliXxX	337	1	1	15
1	darren66	76	9	1	16
0	Ivy33	486	19	0	17
0	Gabrielle	199	12	1	18
1	tr0Y	188	7	1	19
1	bruno	66	1	0	20
0	latisha	218	15	1	21
1	LEO-22	46	1	1	22
0	Sandra42	308	9	0	23
1	Adrian	6	4	0	24
1	drew7	287	9	1	25
0	lenore	395	2	1	26
1	STEpH	118	0	1	27
1	hugh	225	16	1	28
0	Peach	285	7	0	29
0	LUCY	31	9	1	30
1	garrett	249	17	0	31
0	Roxane	322	13	1	32
0	beth4	111	1	1	33
1	parker	419	8	1	34
0	ginn	424	1	0	35
0	-paige-	253	12	1	36

Table 4. Neutral block (1 defeat:1 victory ratio)

oppMale	oppName	oppRank	scoreDiff	win	trial
1	Ned-96	90	2	1	1
1	rick	180	18	0	2
0	Tara	29	19	1	3
0	olivia	261	20	1	4
1	Jens	168	11	0	5
0	-JOY-	88	1	1	6
0	Br00ke	105	2	0	7
1	TraViS	453	10	1	8
0	lacy	338	11	0	9
1	Lex	235	2	0	10
1	jamie55	457	10	1	11
0	Angela	53	11	0	12
1	Stanislas	373	1	0	13
0	letty	369	13	1	14
0	RachelL	281	0	1	15
0	Marylou	93	6	0	16
1	g0nz0	299	11	0	17
1	frederick	150	2	1	18
0	vivian	68	12	0	19
1	Andreww	107	7	0	20
1	MICH404	448	2	1	21
0	Pam369	36	19	0	22
1	Cedric77	122	13	1	23
0	Isabel	27	14	1	24
0	becky	273	2	0	25
1	Matthew	7	15	1	26
1	craig-124	449	17	0	27
1	J8KE	99	10	1	28
0	Caitlin	47	13	1	29
0	Tam111	154	1	0	30
0	Grace	229	17	0	31
1	tomAS	51	2	0	32
0	marsha-	498	1	1	33
1	Gregory	167	20	1	34
1	ffynn	149	16	0	35
0	KIM	32	4	1	36

Table 5. Avatar choices available to players depending on their self-reported gender in the task				
Choice	1	2	3	4
Male				
Female				
Other	