

VI 60 Longcueshock with 1 set of intervals

\--OUTPUTS:

^ACTIVE=1 \SET TO 1 TO EXTEND RETRACTABLE LEVER
 ^feeder=2
 ^PUMP=8
 ^CUELIGHT=4
 ^SPEAKER=6
 ^HOUSELIGHT=5
 ^Shock = 3
 ^Camera=7

^Lever = 1
 ^INACTIVE = 2
 ^FOODCUP = 3

\ List Data Variables Here

\A is the response counter
 \B is the number of rewards
 \C is the timer
 \D is the number of inactive responses
 \E is the number of foodcup responses

\F is the number of responses in the first 5 minutes of the session
 \G is the time (in minutes) when the shocks will occur
 \H is the number of inactive responses in the first 5 minutes of the session
 \I is the number of foodcup responses in the first 5 minutes of the session

\R is the array into which the # of presses per 10 seconds will be placed
 \S is the array into which the # of inactive presses per 10 seconds will be placed
 \W is the array into which the # of foodcup responses per 10 seconds will be placed
 \J is the shock counter
 \K is the shock period (the first 10 sec of the first shock period is 0, the the first shock period of the second shock is 9, the second shock of the second period is 10, etc...)
 \L is the counter that says when data collection in a given shock should stop
 \M is the interval from the beginning of the cue until the shock
 \P is session parameters
 \U is the time a reward will be given
 \V is the list of possible VI ITIs
 \X is the time (in seconds and undisplayable) until the next reward availability
 \Y is the time until the next reward availability
 \Z is the counter for the VI schedule (which vi value it should be)

\O is the array in which the random shock times are stored
 \N is the timer for the random shocks
 \Q is the random shock counter
 \T is the counter for the shock inputs

LIST V = 71,	16,	82,	40,	12,	16,	44,	101,	80,	48,	
57,	40,	49,	82,	108,	45,	5,	38,	56,	115,	61,
102,	91,	85,	11,	31,	43,	30,	16,	45,	95,	20,
111,	113,	96,	44,	39,	37,	70,	92,	2,	79,	27,
60,	85,	58,	65,	28,	70,	52,	101,	31,	75,	117,
55,	42,	111,	39,	13,	77,	118,	83,	35,	1,	54,
92,	48,	75,	5,	116,	29,	96,	63,	115,	5,	28,
108,	36,	28,	18,	30,	45,	76,	26,	107,	1,	77,
117,	104,	27,	113							
LIST O = 554,	120,	793,	63,	643,	79,	750,	169,	590,	196,	
5400										

DIM P = 15 \Session parameters
 DIM G = 10 \Times when shocks will be delivered (in minutes)(this really has 7 slots)
 DIM R = 400 \array for active response data for each time period [formula is
 Page 1

VI 60 Longcueshock with 1 set of intervals

$((g+1)*9)-1$ since there are 9 time periods in every shock presentation) (unless P(2) is changed, then it's $((g+1)*P[2])-1$
DIM S = 400 \array for inactive response data for each time period
DIM W = 400 \array for food cup response data for each time period

\the Gs are the times (in minutes) of the shock events, however, this is somewhat complicated. Here are the rules (well, they aren't rules but they are definitely guidelines)
\1 Make sure that the first trial occurs after the initial context test period (ie: that $G(0) \geq (P(1)/60)$
\2 (and this is really important) Realize that these times are not the times when things will go on in the session but the time from the end of the last trial when things will occur. IE: $G(1) = 10$, $G(2) = 20$ will not start trials at 10 and 20 minutes into the session. It will start trials at 10 minutes into the session and 20 minutes after the end of trial 1 (probably at minute 31 or 32)
\3 Just to drive this home once again, if you have a values of 10, 10, 10 for your intervals and each trial is 2 min long, the trials will occur at 10, 22, and 34 minutes into the session
\4 To really drive this home, there are no checkpoints watching to make sure you type in the right values (for example, to make sure that all of your trials occur within the session), so if you have 2 min trials and 6 intervals of 9 each in a 60 min session, the trials won't occur at 9, 18, 27, 36, 45, and 54; they will occur at 9, 20, 31, 42, 53 and the last one won't occur at all
\5 This one is technically about the trial lengths. Make sure that $P(2)*P(3)=P(4)+P(5)+P(6)$. Otherwise your cues and data collection periods will drift apart after the first trial (i may even put in a test to make sure you do it, but i haven't yet, I'll put a comment in if I do)

DISKVARs = A, B, D, E, F, H, I, K, R, S, W

\ Starts the houselight and run the VI program with VI displays and reward displays (also delivers pellets).

S. S. 1,
S1,

1": SET P(0) = 90; \session length in minutes 7, 2, 1
SET P(1) = 300; \number of seconds in the first data block (in this case, contextual pressing)
SET P(2) = 60; \this is the number of time periods in each shock presentation (if you change this, you need to change DIM H)
SET P(3) = 10; \length of shock data periods in seconds
SET P(4) = 120; \length of precue period
SET P(5) = 360; \length of cue
SET P(6) = 0.5; \length of shock
SET P(7) = 120; \length of post cue period
SET P(10) = 0;
SET P(11) = P(0)+1; \This is used to make sure trials we don't want never happen
SET P(12) = P(2) * P(3);
SET P(13) = P(4) + P(5) + P(7);

SET G(0) = 7; \G(0)time of first shock in minutes
SET G(1) = 5; \this makes sure that no extra trials occur
SET G(2) = 4; \this makes sure that no extra trials occur
SET G(3) = 7; \this makes sure that no extra trials occur
SET G(4) = 6; \this makes sure that no extra trials occur

VI 60 Longcueshock with 1 set of intervals

SET G(5) = P(11);

SET T = 319;

\6,	6,	2,	10,	12,	12,	12,	4,	3,	4	
\9,	2,	9,	4,	10,	8,	5,	11,	5,	9	
\10,	7,	6,	4,	9,	4,	12,	6,	4,	7	
\7,	11,	3,	4,	8,	10,	10,	4,	4,	7	
\day I	10,	2,	8,	10,	5,	4,	6,	11,	11,	5
\day II	5,	2,	2,	8,	11,	9,	11,	2,	11,	7
\day III	8,	2,	4,	12,	5,	3,	12,	6,	9,	9
\day IV	10,	3,	13,	9,	7,	6,	3,	7,	4,	5

IF P(12) = P(13) [@T, @F]

@T: ----> S2

@F: SHOW 1, MATH, 0, 2, DOES, 0, 3, NOT, 0, 4, WORK, 0 ----> STOPKILL

S2,
#START: ON ^HOUSELIGHT; ON ^ACTIVE ----> S3

S3,
0.01": LIST Y = V(Z); SET U = Y + C; SET X = Y*1"; SHOW 1, RF TM, U ----> S4
\ this sets the VI timer

S4,
X#T: ----> S5
\this waits until the time is up and then awaits a response

S5,
#R^Lever: ADD B; SHOW 5, Rewards, B; ON ^feeder; Z1 ----> S3
0.01": SHOW 1, RF TM, 0 ----> SX
\this delivers a reward after a press and it also zeros out the reward timer so you know reward is available

\ Response Counter & Screen Update

S. S. 2,
S1,
#START: SHOW 4, Responses, A, 5, Rewards, B ----> S2

S2,
#R^Lever: ADD A; SHOW 4, Responses, A ----> SX \this is the overall session counter
#R^INACTIVE: ADD D ----> SX
#R^FOODCUP: ADD E; SHOW 3, Foodcup, E ----> SX

\ This runs the switching of data periods during pre, during and post periods of shock
\G is the time (in minutes) when the shocks will occur
\R is the array into which the # of presses per 10 seconds will be placed
\J is the shock counter
\K is the shock period (the first 10 sec of the first shock period is 0, the the first shock period of the second shock is 9, the second shock of the second period is 10, etc...)
\L is the counter that says when data collection in a given shock should stop
\P is session parameters P(2) = the number of time periods in each shock presentation

VI 60 Longcueshock with 1 set of intervals

S. S. 3,

S1,
 1": SET J = 0; SET K = 0; SET L = P(2); SET R(K) = 0; SET G(J) = G(J)*1'; SET
 P(3) = P(3)*1" ----> S2
 \sets the time of the next shock event and sets the number of periods in the next
 shock event

S2,
 #START: ----> S3 \zeros everything out to start and turn the value for period
 length into an actual time

S3,
 G(J)#T: ----> S4 \when the shock event comes, start counting the
 periods

S4,
 P(3)#T: ADD K; SET R(K) = 0; \move to the
 next shock period after the time for the present period ends
 IF K = L [@T, @F]
 @T: ADD J; SET G(J) = G(J)*1'; SET L = L + P(2); SET R(K) = 0; Z2 ----> S3
 \if it is the end of the shock event, reset the shock timer for the next shock
 @F: ADD P(10) ----> SX \if there are
 shock periods left, keep going

\length of precue period SET P(4) = 30;
 \length of cue SET P(5) = 30;
 \length of shock SET P(6) = 0.5;
 \length of post cue period SET P(7) = 30;

S. S. 4,

S1,
 1": SET M = (P(5) - P(6))*1"; SET P(4) = P(4)*1"; SET P(6) = P(6)*1"; SET P(7) =
 P(7)*1" ----> S2
 \sets up time from beginning of cue until shock

S2,
 #START: ----> S3

S3,
 G(J)#T: ----> S4 \when the shock event comes, start counting the
 periods

S4,
 P(4)#T: ON ^SPEAKER ----> S5

S5,
 M#T: ----> S6

S6,
 P(6)#T: OFF ^SPEAKER;
 IF P(7) > 0 [@T, @F]
 @T: ----> S7
 @F: ----> S3

S7,
 P(7)#T: ----> S3

\ This is the part that add counts to the shock period counter
 S. S. 5,

VI 60 Longcueshock with 1 set of intervals

S1,
#START: ----> S2

S2,
G(J)#T: ----> S3 \when the shock event comes, start recording
counts for each period

S3,
#R^Lever: ADD R(K) ----> SX \This adds counts to the counter for each
period an each response (advanced in S. S. 3)
#R^INACTIVE: ADD S(K) ----> SX
#R^FOODCUP: ADD W(K) ----> SX
#Z2: ----> S2 \closes out the counter if it's the end of the
shock period

\This is meant to store the data for the first 5 minutes (or whatever value P(1) is
given)
S. S. 6,

S1,
#START: SET F = 0; SET P(1) = P(1)*1" ----> S2

S2,
P(1)#T: ----> S3
#R^Lever: ADD F ----> SX
#R^INACTIVE: ADD H ---->SX
#R^FOODCUP: ADD I ----> SX

S3,
3000": ----> SX \This doesn't mean anything, just a way to keep the program in
stasis

\ PELLET CONTROL TIMER
\ (Includes K1-pulse for delivering a free pellet)

S. S. 7,
S1,
#Z1: ----> S2

S2,
0.05": OFF ^feeder ----> S1

S. S. 8, \ Increment time "C" with resolution 1 seconds and show it in the program

S1,
#START: ----> S2

S2,
1": ADD C; SHOW 2, Present, C ----> SX

S. S. 9, \ Random Shock Delivery

S1,
1": SET Q = 0; SET N = 0(Q); SET N = N*1" ----> S2 \ 'oh' (Q), not zero(Q)
Page 5

VI 60 Longcueshock with 1 set of intervals
\sets the time of the next random shock event

S2,
#START: ----> S3

S3,
N#T: Z7 ----> S4 \when the shock event comes, start counting the periods

S4,
29.5": ON ^Shock ----> S5

S5,
0.5": OFF ^Shock ----> S6

S6,
30": SET Q = Q+1; SET N = 0(Q); SET N = N*1" ----> S3 \this is an 'oh', not a zero

S. S. 10,
S1,
#Z7: ADD T ---->S2

S2,
10": ADD T ----> S3
#R^Lever: ADD R(T) ----> SX

S3,
10": ADD T ----> S4
#R^Lever: ADD R(T) ----> SX

S4,
10": ADD T ----> S5
#R^Lever: ADD R(T) ----> SX

S5,
10": ADD T ----> S6
#R^Lever: ADD R(T) ----> SX

S6,
10": ADD T ----> S7
#R^Lever: ADD R(T) ----> SX

S7,
10": ----> S1
#R^Lever: ADD R(T) ----> SX

S. S. 11,

S1,
1": SHOW 1, W, 0 ---->S2

S2,
#START: SET P(0) = P(0)*1' ----> S3 \this set the time for the end of the session, W stands for worries, and we have none, actually it is a diagnostic thing to make sure all the variables load before starting the program

S3,
P(0)#T: OFF ^HOUSELIGHT; OFF ^ACTIVE; OFF ^SPEAKER; OFF ^CUELIGHT ---->
STOPABORTFLUSH