

E Prime Notes – Writing an Experiment

The following is a step-by-step guide to building our C1 Pilot Experiment. We used checkerboards placed in five locations to study how the C1 functions. This guide will take you through the process of building our experiment in E-Prime, which also functions as a tutorial to many of the basic tools of E-Prime.

- Trial Procedure
 - Fixation – 100ms
 - ITI – 500 + 0-200ms
 - Stimulus – 50ms
 - ITI – 500 + 0-200ms
 - Task – 200ms
 - Response interval - wait
 - Blink phase – 700ms
- On Startup it asks for:
 - Subject
 - Session
 - Age
 - Sex
 - Handedness
 - Researcher ID
 - Confirm
 - (Override warning, if any)

1. Additional notes

- a. Not mentioned in the steps below, but most text was set to 40 point font and black
 - i. Should be easy enough to figure out how to change stuff like that
- b. Not mentioned in the steps below, but the way to set logging is to go to the logging page on any property pages window and check the boxes you want to record, which is then sent to the DataAid file. Alternatively, in the Duration/Input tab, there are some pre-set groups of data logging settings you can click.
- c. “Gray” color is (128,128,128) RGB values
- a. Ctrl+S to save, it is recommended to generate script* when you save (in the bar second from top, left of run)
 - i. *This will cause an error if done before TrialList is created (see step 6)
 - ii. By generating the script before you save, you have the chance to reload the project from the last save if the script has an error that you don’t know how to fix
 - iii. I learned this the hard way ☹

- b. **Ctrl+Alt+Shift to abort an experiment mid-run**
 - c. BE CAREFUL when COPYING, as E-Studio may (in some cases) link the copy with the original
 - i. (i.e. what you change in the settings of one happens to the other)
 - d. DO NOT NAME ANYTHING AFTER RESERVED WORDS IN EPRIME
 - i. Such as "Continue" "Dim" "Sub" "If" etc
 - ii. I learned this the hard way too ☹
 - e. Data is saved in a DataAid file in the same folder as your experiment
2. Getting started
- a. Turn on PC
 - b. Open E-Studio, create Blank (Professional)
 - c. Double click Experiment Object
 - i. Click on Packages tab
 - 1. Click "Add..." -> Click NetStation -> OK
 - ii. Click on Devices tab
 - 1. Click "Add...", add SRBOX -> OK
 - 2. Click on Display
 - a. Set width to 1920
 - b. Set height to 1080
 - c. Set color bit depth to 32
 - iii. Click Startup Info Tab
 - 1. Check Age, Sex, Handedness, and ResearcherID checkboxes (NetStat. requires)
 - 2. Click ? icon next to each of them to enable
 - 3. Click OK
 - d. Open SessionProc
 - i. Drag PackageCall to beginning of timeline, double click
 - 1. Common tab -> Name: NSInit
 - 2. General tab -> Package: NetStation, Routine: Init
 - 3. Parameters: c, "on", TrialList, "socket", "10.10.10.42"
 - ii. Drag PackageCall to end of timeline, double click
 - 1. Common tab -> Name: NSUnInit
 - 2. General tab -> Package: NetStation, Routine: UnInit
 - 3. Parameters: c
3. Adding instructions that require input to proceed
- a. Open SessionProc
 - b. Drag TextDisplay object to timeline, rename "Instructions" (F2 or right click -> Rename)
 - c. Double click TextDisplay, click "Property Pages" in upper left
 - d. Add desired text, optionally change alignment settings (default center for both)
 - e. Click Duration/Input tab, change duration to (infinite)
 - f. Click "Add..." by input device(s) -> SRBOX -> OK

- i. Note: Default settings are “press any key to terminate,” see right side of Duration/Input tab, “Response Options”
 - g. Timeline should now be NSInit -> Instructions -> NSUninit
- 4. Make the Block of Trials
 - a. Drag List object between Instructions and NSUninit
 - b. Rename it “BlockList”
 - c. Double click to open, click “Add Attribute”
 - i. name “PracticeMode” Default Value “no” -> Add
 - d. Procedure column -> type “PracProc”, enter
 - i. click Yes you want to create it
 - ii. click No it should not be default
 - iii. Change attribute “PracticeMode” to yes
 - e. Click “Add level”, set the Procedure to “BlockProc”
 - i. Yes create it
 - ii. “PracticeMode” = “no”
 - f. Close window, save, take a breath, congratulate self on progress
- 5. Recording from NetStation
 - a. Open BlockProc from Structure window
 - b. Drag PackageCall object to the BlockProc timeline, rename ‘NSStartRecording’
 - i. General tab -> Package: NetStation, Routine: StartRecording, Parameters: c
 - c. Drag PackageCall object to the end of the BlockProc timeline, rename ‘NSStopRecording’
 - i. General tab -> Package: NetStation, Routine: StopRecording, Parameters: c
- 6. Make the List of Trials
 - a. Open BlockProc from Structure window
 - b. Drag TextDisplay object to the beginning of the BlockProc timeline, rename “Cont” (NOT “Continue”)
 - i. Set text: “You will now begin the experiment.” with a prompt to press a button to continue
 - c. Click Duration/Input tab, change duration to (infinite)
 - d. Click “Add...” by input device(s) -> SRBOX -> OK
 - i. Note: Default settings are “press any key to terminate,” see right side of Duration/Input tab, “Response Options”
 - e. Drag List object to the BlockProc timeline between the Start and Stop of Recording
 - i. Rename the object “TrialList”
 - ii. The timeline should now be Cont -> StartRecord -> TrialList -> StopRecord
 - f. Double click TrialList
 - i. Click Add Multiple Attributes
 - 1. Enter 6
 - a. Enter the following data for the two new attributes:
 - b. Name: PositionOfCheckerboard, Default Value: FX
 - c. Name: PositionOfTask, Default Value: FX
 - d. Name: TaskX, Default 50%

- e. Name: TaskY, Default 50%
 - f. Name: CellNumber, Default 1
 - g. Name: CellLabel, Default FX
 - ii. Type into the Procedure column, "TrialProc"
 - 1. Click yes to create it, click yes to make it default
 - iii. Click Add Multiple Levels, enter 24
 - 1. Edit the Positions of Checkerboards and Tasks so that each position of the checkerboard is listed 5 times and it is matched with the 5 positions in the PositionOfTask Column.
 - a. 5 checkerboard positions are FX, UL, UR, LL, LR
 - b. 5 Task positions are FX, U, D, L, R
 - c. Make it an order you can easily assess
 - d. PositionOfTask determines TaskX and TaskY
 - i. FX = 50%, 50%
 - ii. U = 50%, 25%
 - iii. D = 50%, 75%
 - iv. L = 25%, 50%
 - v. R = 75%, 50%
 - e. CellNumber: Correlated to PositionOfCheckerboard
 - i. When PoC = FX, CN = 1
 - ii. UL = 2
 - iii. UR = 3
 - iv. LL = 4
 - v. LR = 5
 - f. CellLabel is identical to PositionOfCheckerboard column
 - iv. Click Property Pages
 - 1. -> Selection tab -> Order = Random
 - 2. -> Reset/Exit tab -> Exit List After 20 cycles (500 samples)
 - g. Close window, generate script, save, take a breath, congratulate self
- 7. Creating the Trial – Fixation
 - a. Double click on TrialProc from the Structure window
 - b. Drag a TextDisplay to the beginning of the timeline, rename it "Fixation"
 - i. Set the text to "+"
 - ii. General tab -> set the forecolor to white and the back color to gray
 - iii. Duration/Input tab -> set the duration to 100ms
- 8. Creating the Trial – Pre Stimulus Jitter
 - a. Drag an Inline object to the end of the timeline, rename it "PreJitter"
 - i. Type the following line: `c.setAttrib "JitDur", 250+ Random(0,200)`
 - b. Drag a TextDisplay to the timeline, rename it "PreJitter"
 - i. Set the text to ""
 - ii. General tab -> set the back color to gray
 - iii. Duration/Input tab -> set the duration to [JitDur]

- iv. Close and return to the TrialProc timeline
- 9. Creating the Trial – Aligning the Refresh Rate
 - a. It is very important for precise timing that a fixed blank interval is placed before the stimulus
 - b. Drag a TextDisplay to the end of the timeline, rename it “Blank”
 - i. Set text to “ ”
 - ii. Gen tab -> set back to gray
 - iii. Dur/In tab -> set duration to 250
- 10. Creating the Trial – Stimulus Display
 - a. Drag an ImageDisplay object to the end of the TrialProc timeline, rename it “Stimulus”
 - i. General tab -> Filename: “STIMULUS_[PositionOfCheckerboard].tif”
 - ii. General tab -> BackColor -> gray
 - iii. Duration/Input tab -> Duration 50ms
 - 1. Data Logging: Time Audit Only
 - 2. PreRelease: 100
- 11. Creating the Trial – Sending the event to NetStation
 - a. Drag an Inline object to the end of the timeline, rename “SendStimEvent”
 - 1. Type: NetStation_SendTrialEvent c, Stimulus
 - ii. Note that it sends the onset of the stimulus
- 12. Creating the Trial – Post Stimulus Jitter
 - a. Drag an Inline object to the end of the timeline, rename it “PostJitter”
 - i. Type the following line: `c.setAttrib “JitDur”,500 + Random(0,200)`
 - b. Drag a TextDisplay to the timeline, rename it “PostJitter”
 - i. Set the text to “ ”
 - ii. General tab -> set the back color to gray
 - iii. Duration/Input tab -> set the duration to [JitDur]
 - iv. Close and return to the TrialProc timeline
- 13. Creating the Trial – Adding the Task
 - a. Drag an List object to the end of the TrialProc timeline, rename it “Task”
 - i. Procedure = “TaskProc”, yes create, yes default
 - ii. Add Attribute -> “Letter”, Default “A”
 - iii. Add multiple levels = 3
 - iv. Set the 4 letters to A, B, C, D respectively
 - v. Property pages
 - 1. Common tab -> Tag = [PositionOfTask]
 - 2. Selection tab -> order = random with replacement
 - 3. Reset/Exit -> reset after 1 sample, exit after 1 sample
 - vi. Open TaskProc timeline, add TextDisplay object, rename “TaskDisplay”
 - vii. General tab -> back color = gray, fore color = white
 - viii. Text = [Letter]
 - ix. Duration/Input tab -> Duration = 200
 - x. Duration/Input tab -> Data logging = Standard

- xi. Frame tab -> Position X = [TaskX], Position Y = [TaskY]

14. Creating the Trial – Adding the Task Response Phase

- a. Drag a List object to the end of the TrialProc timeline, rename it “TaskResponse”
 - i. Procedure: “ResponseProc” yes create, yes default
 - ii. Add multiple attributes: 4
 - 1. FirstLetter, default A
 - 2. SecondLetter, default B
 - 3. ThirdLetter, default C
 - 4. FourthLetter, default D
 - iii. Add multiple levels: 23
 - iv. Edit the attributes columns so that every order is present (A, B, C, D, every letter used once)
 - 1. Be systematic about the combinations
 - v. Selection tab -> order random with replacement
 - vi. Reset/Exit tab -> Reset after 1 sample, exit after 1 sample
 - vii. Open ReponseProc timeline, add TextDisplay, rename “Response”
 - 1. Set text: “([FirstLetter]) ([SecondLetter]) ([ThirdLetter]) ([FourthLetter])”
 - 2. Fore color white, back color gray
 - 3. Duration/Input tab
 - a. Duration infinite
 - b. Data logging: Standard
 - c. Add device -> SRBox

15. Creating the Trial – Sending Response event to NetStation

- a. Drag an Inline object BEFORE the Response on the timeline
 - i. Rename it “SendRespEvent”
 - ii. Type the following lines:
 - 1. NetStation_SendTrialEvent c, Response

16. Creating the Trial – Sending events to NetStation

- a. Drag an Inline object after the BlinkPhase on the timeline
 - i. Rename it “SendEvents”
 - ii. Type the following lines:
 - 1. NetStation_SendTrialEvent c, TaskDisplay
 - 2. NetStation_SendTrialEvent c, Fixation
 - iii. Note that it sends the onset
 - iv. You may want more or less events to be sent, use this as a guide to how to send events
 - 1. The other type of event to send is SendTRSPEvent (TRial SPecific)

17. Creating the Trial – Defining the Trial to NetStation

- a. Add a PackageCall object to the beginning of the TrialProc timeline
 - i. Rename it “NSTrialBegin”
 - ii. General tab -> Package: NetStation, Routine: TrialBegin, Parameters: c
- b. Add a PackageCall object to the end of the TrialProc timeline

- i. Rename it “NSTrialEnd”
 - ii. General tab -> Package: NetStation, Routine: TrialEnd, Parameters: c
- 18. Add a Practice Mode
 - a. Drag a List object to the PracProc timeline, rename it “PraList”
 - i. Make the PraList identical to TrialList on the page with the excel table
 - 1. (**Weight = 1 instead of 20**, Procedure TrialProc, PositionOfCheckerBoard, PositionOfTask, TaskX, TaskY, CellNumber, CellLabel)
 - ii. Property pages -> Selection tab -> order random
 - iii. Reset/Exit tab -> Reset after 20 samples, Exit after 20 samples
- 19. Generate script, Save, and Run
 - a. Check if all timelines are in order (see below)
 - b. There are no errors in the script
 - c. Your experiment runs as you would like
 - d. It logs the information you want to send to NetStation (super important)
 - e. It logs the information you want to send to the DataAid file
 - f. Generate script, save, and run it!
 - g. Congratulations, you have made an experiment!

SESSION TIMELINE

- 1. SessionProc
 - a. NSInit
 - b. Instructions
 - c. BlockList
 - i. PracProc
 - 1. PraList
 - a. TrialProc
 - i. NSTrialBegin
 - ii. Fixation
 - iii. SetPreJitter
 - iv. PreJitter
 - v. Blank
 - vi. Stimulus
 - vii. SendStimEvent
 - viii. SetPostJitter
 - ix. PostJitter
 - x. Task
 - 1. TaskProc
 - a. TaskDisplay
 - xi. TaskResponse

