2021 ■ Vol. 17 ■ no. 3

The Corsi Blocks Task: Variations and coding with jsPsych

Rose-Marie Gibeau [a]

[a]University of Ottawa

Abstract ■ The Corsi Blocks Task is a widely used task in both clinical and experimental work. However, many versions exist, and it can be difficult to, firstly, choose between the variations of each parameter and, secondly, to program it in a computer software. No article has yet been published on the different versions of the computerized Corsi Blocks Task. Herein, we summarize possible variations of this task. We also provide an implementation of the task using jsPsych.

Keywords ■ Corsi Block Task, spatial working memory task, jsPsych, research methods. **Tools** ■ JavaScript, jsPsych.

Acting Editor ■ Denis Cousineau (Université d'Ottawa)

rgibe089@uottawa.ca

10.20982/tqmp.17.3.p299

Introduction

The Corsi Blocks Task, also called Corsi Block-Tapping Task, is wildly used since its publication (Berch, Krikorian, & Huha, 1998). The task was created in 1970 by P. M. Corsi as part of his thesis dissertation (Corsi, 1972) to examine the effect of hippocampal lesions on memory (Milner, 1971). Although it was elaborated for clinical research, it has since been used in various domains of study (Berch et al., 1998), for example, short-term memory (Vecchi & Richardson, 2001; Vandrierendonck, Kemps, Fastame, & Szmalec, 2010), attention (Smyth & Scholey, 1994), memory deficits (De Renzi & Nichelli, 1975) and spatial abilities (Capitani, Laiacona, & Ciceri, 1991).

The original version of the Corsi Blocks Task presented nine 3-cm wooden blocks placed in a unique disposition on a 20 cm × 25 cm wooden board, as shown in Figure 1, left panel (Milner, 1971; Furley & Memmert, 2010; Hamidi, Azizolahi, Rasti, & Beigi, 2020). During one trial, the researcher taps some of the blocks in a particular sequence and the participants are required to repeat that sequence in the correct order. No information is given regarding the length of the sequences, but the original version of the task has the following specific procedure. First, the participant's spatial span (i.e., the longest sequence length the participant can recall correctly) is assessed. Then, 24 sequences, one item longer than the spatial span, is presented. Unbeknownst to the participant, every third sequence (i.e., third, 6th, 9th, etc.) is constant, meaning that the same sequence is repeated.

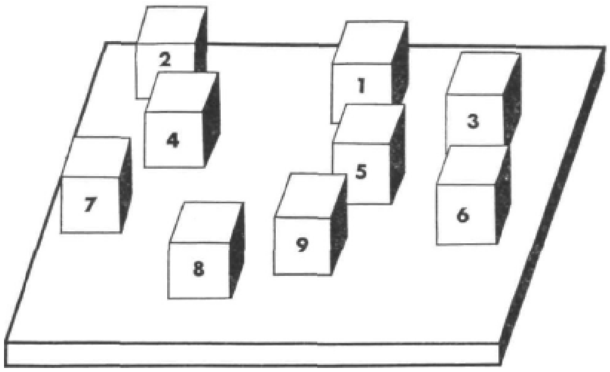
With the advent of computers, many working memory tasks are being adapted for online use (Shackman et al., 2006; Redick et al., 2012). The computerized version of the Corsi Blocks Task has been used in studies since the late 1980s (starting with Morris et al., 1988; Smyth & Scholey, 1992). With this computerization comes many possibilities concerning the parameters of the task. Reviewing the literature, we found a large number of variations in terms of color, sequence lengths, rate of presentation, etc. In fact, almost all articles examined herein used a unique variation of the computerized task. Clearly, there is no standard computerized Corsi Blocks Task. This is problematic because it makes replication difficult, and in case of disagreement between studies, finding the cause is more hazardous.

For the traditional (non-computerized) task, Kessels, van Zandvoort, Postma, Kappelle, and de Haan (2000) detailed the standardized version of the task presented by Corsi (1972). However, because no work has been published on standardizing the computerized version, it is difficult to decide on which variation to use, especially for those who are not familiar with computer programing.

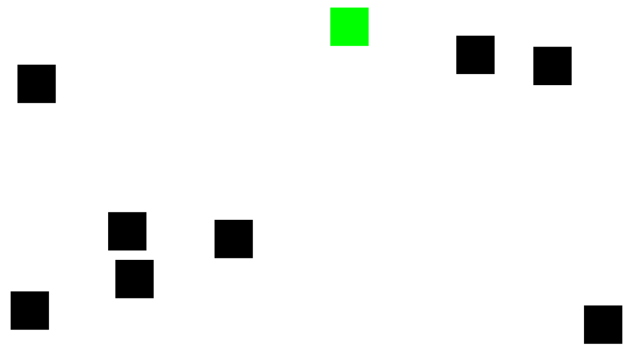
The first goal of this text is to describe the different variations of the task, with references for these parameters. The second goal is to present a program in jsPsych (de Leeuw, 2015) where all the parameters can be set explicitly

Figure 1 ■ Left: the original Corsi Blocks task consisted of wooden blocks disposed on a board, here as seen from the experimenter's viewpoint (participants do not see each block's assigned number). The figure is taken from Milner (1971). Right: the computerized version of the Corsi Blocks task in which one block is currently highlighted.

(a)



(b)



without programming skills.

Methodological variations

In the computerized versions, 3D blocks are replaced by 2D squares. Instead of being tapped upon by the experimenter, they are highlighted in turn.

Colors

Whereas the traditional version used black or brown wooden blocks, the computerized versions use squares of a variety of colors. While in some articles, the color is not mentioned (Stoffers, Berendse, Deijen, & Wolters, 2003; Visser et al., 2019; Athar, Atef-Vahid, & Ashouri, 2020), it is mentioned in most others. Some articles used blue squares that change to red when they are highlighted and when they are selected by the participant (Kennedy et al., 2020; Kennedy et al., 2017; Kennedy, Wightman, Khan, Grothe, & Jackson, 2018). In one article, blue squares turn yellow when they are highlighted (Nagy, Kalmár, Mária Beke, Gráf, & Horváth, 2019). Others use grey squares that are highlighted using red or white colors (Lancia, Cofini, Carrieri, Ferrari, & Quarestima, 2018; White, Forsyth, Lee, & Machado, 2018). In most articles, however, white squares are displayed (Morris et al., 1988; Joyce & Robbins, 1991; Smyth & Scholey, 1992, 1994; Visu-Petra, Benga, Tincas, & Miclea, 2007; Higo, Minamoto, Ikeda, & Osaka, 2014; Vandrierendonck et al., 2010). In one of these studies, the white squares are highlighted using the color blue (Morris et al., 1988), while in the others the squares are highlighted using black. Lastly, one study displayed dark yellow squares that changed to bright yellow when they were selected by the participant (Claessen, van der Ham, & van Zandvoort, 2015).

The background is typically one of three colors: blue (Higo et al., 2014; Vandrierendonck et al., 2010), black (Lancia et al., 2018; White et al., 2018), or white (Smyth & Scholey, 1992, 1994). In studies presenting white squares on a white background, the squares are outlined in black (Smyth & Scholey, 1992, 1994).

Highlighting the sequence

To highlight the squares, researchers typically change the block colors for a short period of time, so the squares don't blink. However, one study presented black squares in which an "X" appeared in the middle when highlighted (Fischer, 2001). The color of the X is not mentioned but may be white considering the squares are black. Smith-Spark and Fisk (2007) also used Xs instead of changing the color to highlight the squares.

For computer presentation where responses will be given with the mouse, one study mentioned that the mouse pointer is visible but fixed into place while the sequence of to-be-remembered items is presented (Kennedy et al., 2018; Kennedy et al., 2017; Kennedy et al., 2020). Most studies do not provide that information.

Size

In the computerized versions, there is usually little information about the size of the squares and the size of the screen. Out of eight articles where this information was provided, three presented 25 mm × 25 mm squares on a 42 cm, a 38 cm, or a 30 cm computer screen (Higo et al., 2014; Vandrierendonck et al., 2010; Joyce & Robbins, 1991, respectively). Claessen et al. (2015) used a 22 mm × 22 mm on

Table 1 ■ Item parameters

Variable	Purpose
`OneItemShape`	The shape to be shown using scalable vector graphic protocol. "rect" is the primitive for rectangles
`OneItemShownColor`	The color of the items when initial shown. "`rgb(0,0,0)`" is the red-green-blue values for black squares
`OneItemBlinkColor`	"`rgb(0,255,0)`" is green
`OneItemClickColor`	"`rgb(126,126,126)`" is gray
`OneItemSize`	The size of the item as a fraction of the smallest dimension (height or width) of the screen. For example, 1/10 (one-tenth of the screen's height) represents 105 pixels on a 1680 × 1050 computer screen)
`OneItemMargin`	Margin to be left blank around items as a fraction of the screen size. For example, 1/50 means 21 pixels on the same screen as above.

Note. The size should be written relative to the display size, not in pixels, because some computers have screens with lower resolutions.

Table 2 ■ Trial parameters

Variable	Purpose
`InterTrialDuration`	Duration between each trial. For example, 2000 ms.
`FixationDuration`	Duration of the fixation cross. For example, 400 ms.
`MouseToBeHidden`	Is the computer mouse hidden during the trial. For example, Yes!
`PreBlinkDuration`	Duration before the item flash. For example, 1000 ms.
`BlinkDuration`	Duration where the item is highlighted. For example, 400 ms.
`PostBlinkDuration`	Duration before recall. For example, 1000 ms.
`InterBlinkDuration`	Duration between the end of a blink and the start of the next. For example, 400 ms.
`RecallSignal`	The sound that signals the start of recall. For example, 500 hz for 400 ms.
`RecallSignalDuration`	Duration of the sound. For example, 400 ms.
`AcknowledgeDuration`	Duration where the clicked button change format. For example, 200 ms.

a 20 cm × 14 cm iPad screen. One study used the same size as in the traditional version (20 cm × 25 cm; Lancia et al., 2018). Fischer (2001) presented 28 mm side-length squares. The seventh and eighth articles used two different array sizes (Smyth & Scholey, 1992, 1994). In the large-squares array, 44 mm squares were displayed on a 53 cm screen, and in the small-squares array, 22 mm squares were displayed on the same screen.

It is rarely indicated if the computer screen had a ratio height to width of 16:9 or 4:3; the former tending to be the standard past 2010.

Rate of presentation

The rate of presentation is relatively similar across studies. Indeed, most of the articles that specify the rate indicated that the squares are highlighted for 1000 ms, some with 1000 ms inter-item interval, others, with no interval (Fischer, 2001; Vecchi & Richardson, 2001; Higo et al., 2014; Lancia et al., 2018; White et al., 2018). Claessen et al. (2015) highlighted the squares for 500 ms with an inter-item delay of 1000 ms. Vandrierendonck et al. (2010) had a 500 ms inter-item delay. Other studies highlighted the squares for either 3000 ms or 1500 ms (Joyce & Robbins, 1991; Smyth & Scholey, 1992, 1994, respectively).

Inter-trial separation

Some studies added other cues for their participants. Specifically, Vandrierendonck et al. (2010) had a 400 ms-sound at the start of each trial and an inter-trial interval of 2000 ms. White et al. (2018) also presented a sound after each sequence for 300 ms. Higo et al. (2014), unlike others, had a three-second countdown at the beginning of each trial and a fixation cross that disappeared after 2000 ms.

Sequence Length and repetitions

Almost all of the articles reviewed started the task using a sequence length of two or three squares, except two (both started with a one-square sequence; Morris et al., 1988; White et al., 2018). Most of them had a maximum sequence length of nine. Others had a maximum length of eight, seven, or six (e.g., Athar et al., 2020; Smyth & Scholey, 1994;

Table 3 ■ Experiment parameters

Variable	Purpose
`BackgroundColor`	the color of the background of the screen
`TextColor`	the color of the text (instructions, fixation point, welcome text and bye)
`NItems`	The number of total items. For example, 9 in the original Corsi Blocks task.
`sequenceLengths`	The sequence lengths tested. For example, a sequence of 3, 5 or 7 squares.
`sequenceRandomOrder`	Whether the sequence lengths are tested in a random order across participants or in the order given in sequenceLengths
`sequenceRepetition`	How many trials there is within a given sequence length.

Lancia et al., 2018, respectively). One study had three levels of difficulty (three, five, and seven squares; Higo et al., 2014). Across the studies, each sequence length is repeated between two to six times, with the most common being two and three times. These lengths are presented in increasing order only, or randomly intermixed.

Total number of squares

Every study examined had nine squares as in the traditional task, except one that had 12 different positions (Smith-Spark & Fisk, 2007).

Square positions

Most of the studies that describe the square positions used the display of the original task (Corsi, 1972; Milner, 1971). Fischer (2001) used the position of Schellig and Hättig (1993, their Figure 3). However, Kennedy et al. (2017, 2018, 2020) randomly positioned the squares on the screen with the only constraints being that the squares could not overlap.

Responding

In all studies, at the end of each trial, participants had to click or tap (touch) on the squares they think were highlighted. For most studies, when the participant did so, the square changed color as mentioned earlier. Some studies did not mention what happened (if something happened) when participants clicked or tapped a square; e.g., Fischer (2001).

Dependent variables measured

Ghavidel, Fadardi, Gatto, Sedeghat, and Tabibi (2020) reported three ways to score the Corsi Blocks Task. Most researchers use the longest sequence recalled without error. In some variations, participants must report the sequence in the forward order or backward (reversed) order (Berch et al., 1998). Thus, some studies compute the immediate block forward span divided by the immediate block backward span (Ghavidel et al., 2020). The third way to score the task is to report the total number of correctly recalled sequences.

Programming

For a study currently in preparation, the computerized Corsi Blocks Task is administered as a measure of spatial working memory. The computer program used for the study in question is presented in the present article. In Appendix A, simple and easily changeable code lines are presented. The code is also available on this journal's website. Tables 1, 2, and 3 present the relevant parameters as found in the code, while Algoritms 1 and 2 at the end show the sequence of events at the trial level and the experiment level.

The code was programmed with a free and open-source suite extending javascript (de Leeuw, 2015). It works as a stand-alone application from any web browser. It can also be adapted to work within Qualtrics (Choe, 2020). The stand-alone version is found on the journal's website.

Conclusion

To summarize, The Corsi Blocks Task is a useful measure in both clinical and experimental research. Considering the advancement of technology and life circumstances (e.g., COVID-19 pandemic), more tasks are being computerized. Unfortunately, studies using the computerized version of the Corsi Blocks Task use unique parameter variations, as seen in the present text. As such, those variations were described, and their code lines are included to enable easy task programming. However, no formal validation of this task has been done. No study has assembled the evidence, accumulated through the years, for or against its validity and reliability. Whether it is a measure of spatial memory span or something else is still an open question.

References

Athar, M. E., Atef-Vahid, M.-K., & Ashouri, A. (2020). The influence of shift work on the quality of sleep and executive functions. *Journal of Circadian Rhythms*, *18*(1), 1–8. doi:10.5334/jcr.194

Berch, D. B., Krikorian, R., & Huha, E. M. (1998). The Corsi Block-Tapping Task: Methodological and theoretical considerations. *Brain and Cognition*, *38*(3), 317–338. doi:10.1006/brcg.1998.1039

Capitani, E., Laiacona, M., & Ciceri, E. (1991). Sex differences in spatial memory: A reanalysis of block tapping long-term memory according to the short-term memory level. *The Italian Journal of Neurological Sciences*, *12*, 461–466.

Choe, K. W. (2020). jsPsych in Qualtrics tutorial series. Retrieved August 6, 2021, from https://kywch.github.io/jsPsych-in-Qualtrics/

Claessen, M. H. G., van der Ham, I. J. M., & van Zandvoort, M. E. E. (2015). Computerization of the standard Corsi Block-Tapping Task affects its underlying cognitive concepts: A pilot study. *Applied Neuropsychology: Adult*, *22*(3), 180–188. doi:10.1080/23279095.2014.892488

Corsi, P. M. (1972). *Human memory and the medial temporal region of the brain* (Doctoral dissertation, McGill University).

De Renzi, E., & Nichelli, P. (1975). Verbal and nonverbal short-term memory impairment following hemispheric damage. *Cortex*, *11*(4), 341–354. doi:10.1016/S0010-9452(75)80026-8

de Leeuw, J. R. (2015). jsPsych : A JavaScript library for creating behavioral experiments in a web browser. *Behavior Research Methods*, *47*, 1–12. doi:10.3758/s13428-014-0458-y

Fischer, M. H. (2001). Probing spatial working memory with the Corsi Blocks task. *Brain and Cognition*, *45*(2), 143–154. doi:10.1006/brcg.2000.1221

Furley, P., & Memmert, D. (2010). Differences in spatial working memory as a function of team sports expertise: The Corsi Block-Tapping task in sport psychological assessment. *Perceptual and Motor Skills*, *110*(3), 801–808. doi:10.2466/pms.110.3.801-808

Ghavidel, F., Fadardi, J. S., Gatto, N. M., Sedeghat, F., & Tabibi, Z. (2020). Feasibility of using a computer-assisted working memory training program for healthy older women. *Cognitive Processing*, *21*, 383–390. doi:10.1007/s10339-020-00975-7

Hamidi, F., Azizolahi, M., Rasti, J., & Beigi, F. (2020). Effectiveness of computer games on improving the attention and working memory of children with attention deficit hyperactivity disorder. *Health Education and Health Promotion*, *8*(2), 67–72.

Higo, K., Minamoto, T., Ikeda, T., & Osaka, M. (2014). Robust order representation is required for backward recall in the Corsi blocks task. *Frontiers in Psychology*, *5*, 1285–1299. doi:10.3389/fpsyg.2014.01285

Joyce, E. M., & Robbins, T. W. (1991). Frontal lobe function in korsakoff and non-korsakoff alcoholics: Planning and spatial working memory. *Neuropsychologia*, *29*(8), 709–723. doi:10.1016/0028-3932(91)90067-I

Kennedy, D. O., Bonnländer, B., Lang, S. C., Pischel, I., Forster, J., Khan, J., . . . Wightman, E. L. (2020). Acute and chronic effects of green oat (avena sativa) extract on cognitive function and mood during a laboratory stressor in healthy adults: A randomised, double-blind, placebo-controlled study in healthy humans. *Nutrients*, *12*(6), 1–19. doi:10.3390/nu12061598

Kennedy, D. O., Jackson, P. A., Forster, J., Khan, J., Grothe, T., Perrinjaquet-Moccetti, T., & Haskell-Ramsay, C. F. (2017). Acute effects of a wild green-oat (avena sativa) extract on cognitive function in middle-aged adults: A doubleblind, placebo-controlled, within-subjects trial. *Nutritional Neuroscience*, *20*(2), 135–151. doi:10.1080/1028415X.2015.1101304

Kennedy, D. O., Wightman, E., Khan, J., Grothe, T., & Jackson, P. (2018). The acute and chronic cognitive and cerebral blood-flow effects of a sideritis scardica (greek mountain tea) extract—a randomized, double-blind, placebo-controlled study in healthy humans. *Nutrients*, *10*(8), 1–99. doi:10.3390/nu10080955

Kessels, R. P. C., van Zandvoort, M. J. E., Postma, A., Kappelle, L. J., & de Haan, E. H. F. (2000). The Corsi Block-Tapping task: Standardization and normative data. *Applied Neuropsychology*, *7*(4), 252–258. doi:10.1207/S15324826AN0704_8

Lancia, S., Cofini, V., Carrieri, M., Ferrari, M., & Quaresima, V. (2018). Are ventrolateral and dorsolateral prefrontal cortices involved in the computerized Corsi block-tapping test execution? An fNIRS study. *Neurophotonics*, *5*, 1–99. doi:10.1117/1.NPh.5.1.011019

Milner, B. (1971). Interhemispheric differences in the localization of psychological processes in man. *British Medical Bulletin*, *27*(3), 272–277. doi:10.1093/oxfordjournals.bmb.a070866

Morris, R. G., Downes, J. J., Sahakian, B. J., Evenden, J. L., Heald, A., & Robbins, T. W. (1988). Planning and spatial working memory in parkinson's disease. *Journal of Neurology, Neurosurgery & Psychiatry*, *51*(6), 757–766.

Nagy, A., Kalmár, M., Mária Beke, A., Gráf, R., & Horváth, E. (2019). Intelligence and executive functions in 9–10 year-old preterm children born with very low and extremely low birth weight. *Mentálhigiéné és Pszichoszomatika*, *20*(4), 421–440.

Redick, T. S., Broadway, J. M., Meier, M. E., Kuriakose, P. S., Unsworth, N., Kane, M. J., & Engle, R. W. (2012). Measuring working memory capacity with automated complex span tasks. *European Journal of Psychological Assessment*, *28*(3), 164–171. doi:10.1027/1015-5759/a000123

Schellig, D., & Hättig, H. A. (1993). Die bestimmung der visuellen merkspanne mit dem block board. *Zeitschrift für Neuropsychologie*, *4*, 104–112. Retrieved from https://www.researchgate.net/publication/286272982

Shackman, A. J., Sarinopoulos, I., Maxwell, J. S., Pizzagalli, D. A., Lavric, A., & Davidson, R. J. (2006). Anxiety selectively disrupts visuospatial working memory. *Emotion*, *6*(1), 40–61. doi:10.1037/1528-3542.6.1.40

Smith-Spark, J. H., & Fisk, J. E. (2007). Working memory functioning in developmental dyslexia. *Memory*, *15*(1), 34–56. doi:10.1080/09658210601043384

Smyth, M. M., & Scholey, K. A. (1992). Determining spatial span: The role of movement time and articulation rate. *The Quarterly Journal of Experimental Psychology*, *45*(3), 479–501. doi:10.1080/02724989208250624

Smyth, M. M., & Scholey, K. A. (1994). Interference in immediate spatial memory. *Memory and Cognition*, *22*(1), 1–13. doi:10.3758/BF03202756

Stoffers, D., Berendse, H. W., Deijen, J. B., & Wolters, E. C. (2003). Deficits on Corsi's block tapping task in early stage parkinson's disease. *Parkinsonism and Related Disorders*, *10*(2), 107–111. doi:10.1016/S1353-8020(03)00106-8

Vandrierendonck, A., Kemps, E., Fastame, M. C., & Szmalec, A. (2010). Working memory components of the Corsi blocks task. *British Journal of Psychology*, *95*(1), 57–79. doi:10.1348/000712604322779460

Vecchi, T., & Richardson, J. T. E. (2001). Measures of visuospatial short-term memory: The Knox cube imitation test and the Corsi Blocks Test compared. *Brain and Cognition*, *46*(1-2), 291–295. doi:10.1016/S0278-2626(01)80086-5

Visser, T. A. W., Bender, A. D., Bowden, V. K., Black, S. C., Greenwell-Barnden, J., Loft, S., & Lipp, O. V. (2019). Individual differences in higher-level cognitive abilities do not predict overconfidence in complex task performance. *Consciousness and Cognition*, *74*, 1–99. doi:10.1016/j.concog.2019.102777

Visu-Petra, L., Benga, O., Tincas, I., & Miclea, M. (2007). Visual-spatial processing in children and adolescents with down's syndrome: A computerized assessment of memory skills. *Journal of Intellectual Disability Research*, *51*(12), 942–952. doi:10.1111/j.1365-2788.2007.01002.x

White, N., Forsyth, B., Lee, A., & Machado, L. (2018). Repeated computerized cognitive testing: Performance shifts and test–retest reliability in healthy young adults. *Psychological Assessment*, *30*(4), 539–549. doi:10.1037/pas0000503

Algorithm 1 ■ Sequence of events at the experiment level

Run the whole experiment	Parameters
`FullScreenOn`	
`SayWelcome`	until response

`RunOneTrial`	(see Algorithm 2)
	`sequenceLengths`
	`sequenceRepetition`
	`sequenceRandomOrder`
	`nItems`

`FullScreenOff`	
`SayBye`	until response

Algorithm 2 ■ Sequence of events at the trial level

RunOneTrial	Parameters
SetnBlinkingItems	nBlinkingItems is set
GeneratePositions	PositionFunction; ListOfItems, ListOfBlinkingItems, etc., are set
WaitInterTrialDuration	InterTrialDuration
ShowFixation	FixationDuration, mouseToBeHidden
ShowAllItems	ListOfItems
ShowOneItem	PreBlinkDuration
WaitPreBlinkDuration	
BlinkAllItems	ListOfBlinkingItems
BlinkOneItem	BlinkDuration
WaitBlinkDuration	InterBlinkDuration
ShowOneItem	InterBlinkDuration
WaitInterBlinkDuration	
WaitPostBlinkDuration	PostBlinkDuration
StartResponding	
ShowMouse	
ReadAllResponses	
ReadOneResponse	
BlinkThatResponse	
GatherResponses	'call-function'; set RTs, ACC, and responses given

Note: All duration variables are in ms.

Appendix A: Complete listing of the jsPsych code

The following is the complete code to run a computerized Corsi Blocks task in a web browser with both HTML and JavaScript coding. It uses the library of functions provided with jsPsych version 6.1. The only change is a homemade plugin, html-keyboard-response-noerase (where the display is not erased; that feature should be part of jsPsych version 7.0).

In the HTML code, comments are denoted with `<!- comment ->` whereas in the JavaScript sections, comments begin with `// and span the remaining of the line`.

The sound file contains a sound wave of 400 hz lasting 500 ms. In the script, we added calls to `console.log` to display information in the browser's console (typically, F12 will open the browser's console). These instructions can be omitted.

We used the following convention in naming the parameters and the procedures: all the procedures have names beginning with a verb (e.g., `RunTrials`), whereas the parameters have no verb in their names (e.g., `nBlinkingItems`).

Note that the current script does not save the data. Once gathered, it is your responsibility to preserve the data on a server where they can be retrieved at a later moment. To save the data with jsPsych, here are some references.

- Only for jsPsych (especially useful for in-lab experiments): www.jspsych.org/overview/data/ and www.jspsych.org/core_library/jspsych-data/;
- For the jsPsych-Qualtrics duo: kywch.github.io/jsPsych-in-Qualtrics/save-php/;
- For the jsPsych-Pavlovia duo: pavlovia.org/js-psych.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- character encoding -->
    <meta content="text/html;charset=utf-8" http-equiv="Content-Type">
    <meta content="utf-8" http-equiv="encoding">
    <!-- Title of the window -->
    <title>Corsi Blocks task: Gibeau (2021).</title>
    <!-- Gibeau, R.-M. (submitted)                                         -->
```

```
  <!--     The Corsi Blocks task: Variations and coding with jsPsych. -->
  <!--     The Quantitative Methods for Psychology.                   -->

  <!-- declaring the jsPsych functions needed -->
  <!-- note that jspsych-html-keyboard-response-noerase.js IS NOT  -->
  <!-- part of the standard bundle -->
  <script src="jspsych-6.1.0/jspsych.js"></script>
  <script src="jspsych-6.1.0/plugins/jspsych-html-keyboard-response-noerase.js"></script>
  <script src="jspsych-6.1.0/plugins/jspsych-html-keyboard-response.js"></script>
  <script src="jspsych-6.1.0/plugins/jspsych-audio-button-response.js"></script>
  <script src="jspsych-6.1.0/plugins/jspsych-audio-keyboard-response.js"></script>
  <script src="jspsych-6.1.0/plugins/jspsych-audio-keyboard-response.js"></script>
  <script src="jspsych-6.1.0/plugins/jspsych-html-button-response.js"></script>
  <script src="jspsych-6.1.0/plugins/jspsych-fullscreen.js"></script>
  <script src="jspsych-6.1.0/plugins/jspsych-call-function.js"></script>

  <!-- uploading the jsPsych style sheet -->
  <link href="jspsych-6.1.0/css/jspsych.css" rel="stylesheet" type="text/css"></link>
</head>

<body>
  <script>
    // *****************************
    // *   Defining general stuff    *
    // *****************************
    console.log('Begining of the parameter definitions')

    // Experiment parameters
    var BackgroundColor    = "lightblue" // accept the HTML-defined colors
    var TextColor          = "blue"      //
    var nItems              = 7          // total number of squares on the display
    // the sequence lengths to test the participant with, with the variable "OneSequenceLength"
    var sequenceLengths     = [{OneSequenceLength: 3},{OneSequenceLength: 5}];
    var sequenceRandomOrder = true       // true for "random" or false for "sequential"
    var sequenceRepetition  = 2          // how many times each length is tested

    // Trial parameters
    var InterTrialDuration   = 100
    var FixationDuration     = 400
    var MouseToBeHidden      = true
    var PreBlinkDuration     = 500
    var BlinkDuration        = 400
    var PostBlinkDuration    = 500
    var InterBlinkDuration   = 400
    var RecallSignal         = 'https://raw.githubusercontent.com/RMG2424/Dr.-Mid-Nite/master/
  CorsiBlockjsPsych/500hz-400ms.wav'
    var RecallSignalDuration = 400
    var AcknowledgeDuration  = 200

    // Item parameters; assembled using makeItem
    var OneItemShape      = "rect"              // one of the svg primitive
    var OneItemShownColor = "rgb(0,0,0)"        // black
    var OneItemBlinkColor = "rgb(127,127,127)"  // gray
    var OneItemClickColor = "rgb(0,0,255)"      // blue
    var OneItemSize       = 1/10                // proportion of the item relative to minscreen
    var OneItemMargin     = 1/50                // margin to leave empty around the item

    console.log('End of the parameter definitions')

    // ***********************************************
    // *  THIS IS IT! Everything beyond this point    *
    // *   will run nicely from the above definitions *
    // ***********************************************

    // set the background color
    document.body.style.backgroundColor = BackgroundColor;
```

```
  document.body.style.color = TextColor;

  // these two lines convert the relative sizes into pixel sizes LEAVE UNCHANGED
  var OneItemSizePX     = Math.floor(OneItemSize * Math.min(screen.width, screen.height))
  var OneItemMarginPX   = Math.floor(OneItemMargin * Math.min(screen.width, screen.height))
  console.log(`Item dimensions in pixels: (size) ${OneItemSizePX}, (margins) ${OneItemMarginPX}`);

  // Function that builts an HTML svg (scalable vector graphic) image
  // may need to be adapted if a different shape is chosen
  var makeItem = function(color) {
    return(`<svg width="${OneItemSizePX}" height="${OneItemSizePX}"><${OneItemShape} width="${
OneItemSizePX}" height="${OneItemSizePX}" style="fill:${color};stroke-width:0;"></svg>`);
  }

  // Empty lists of items; are to be populated by PositionFunction
  var ListOfItems         = new Array(nItems); // the array is full length
  var ListOfBlinkingItems = new Array(nItems); // idem
  var ListOfButtons       = new Array(nItems);
  var nBlinkingItems      = null;

  // Function that generates non-contiguous positions randomly.
  // This function can place the items anywhere on screen
  // while leaving an empty margin around the items.
  var PositionFunction = function() {
    // console.log("begin of PositionFunction");
    // console.log("nBlinkingItems: ", nBlinkingItems);
    var i, j;
    var x, y;
    var tooclose = true;
    ListOfButtons.length = 0;

    for (i = 0; i < nItems; i++){
      tooclose = true;
      while(tooclose) {
        tooclose = false;
        x = Math.floor(OneItemMarginPX + Math.random() * (screen.width-OneItemSizePX-2*
OneItemMarginPX) );
        y = Math.floor(OneItemMarginPX + Math.random() * (screen.height-OneItemSizePX-2*
OneItemMarginPX) );
        for (j= 0; j <i; j++) {
          if ((x > ListOfItems[j].x - OneItemSizePX - OneItemMarginPX)&&
              (x < ListOfItems[j].x + OneItemSizePX + OneItemMarginPX)&&
              (y > ListOfItems[j].y - OneItemSizePX - OneItemMarginPX)&&
              (y < ListOfItems[j].y + OneItemSizePX + OneItemMarginPX)){
            tooclose = true;
          };
        };
      };
      ListOfItems[i] = [];
      ListOfItems[i].x = x;
      ListOfItems[i].y = y;
      ListOfItems[i].ItemHTML  = `<p style = "position:absolute; margin-top: 0em; left: ${x}px; top:
 ${y}px">`+
                              `${makeItem(OneItemShownColor)} </p>`;
      ListOfItems[i].ItemBLANK = `<p style = "position:absolute; margin-top: 0em; left: ${x}px; top:
 ${y}px">`+
                              `${makeItem(OneItemBlinkColor)}</p>`;
      ListOfItems[i].ItemACKNW = `<p style = "position:absolute; margin-top: 0em; left: ${x}px; top:
 ${y}px">`+
                              `${makeItem(OneItemClickColor)}</p>`;
      // they are also concatenated in a list of buttons
      ListOfButtons.push(`<button type=button style = "position:absolute; `+
            `margin-top: 0px; padding: 0px; background: none; `+
            `border: none; left: ${x}px; top: ${y}px">`+
            `${makeItem(OneItemShownColor)}</button>` );
      if (i< nBlinkingItems) {
```

```
        // the first nBlinkingItems are also kept in a shorter list
        ListOfBlinkingItems[i] = [];
        ListOfBlinkingItems[i].x = x;
        ListOfBlinkingItems[i].y = y;
        ListOfBlinkingItems[i].ItemHTML  = `<p style = "position:absolute; margin-top: 0em; left: ${
x}px; top: ${y}px">${makeItem(OneItemShownColor)}</p>`;
        ListOfBlinkingItems[i].ItemBLANK = `<p style = "position:absolute; margin-top: 0em; left: ${
x}px; top: ${y}px">${makeItem(OneItemBlinkColor)}</p>`;
      }
    }
    // console.log("end of PositionFunction ");"
  }
  console.log('End of the definitions')

  // ***********************************************
  // ********* Defining mouse pointer events *********
  // ***********************************************

  // mouse pointer functions
  var HideMouse = function() { if (MouseToBeHidden) {
      document.querySelector('head').insertAdjacentHTML('beforeend', '<style id="cursor-toggle">
html { cursor: none; } </style>');
    }
  }
  var ShowMouse = function() { if (MouseToBeHidden) {
      document.querySelector('#cursor-toggle').remove();
    }
  }

  // **************************************
  // *   Defining experiment-level events   *
  // **************************************

  // Toggle full screen on or off
  var FullScreenOn = {
    type: 'fullscreen',
    message: "<p>The experiment will be in full screen mode once you click on the button.</p>",
    button_label: 'Full Screen Mode',
    fullscreen_mode: true
  }
  var FullScreenOff = {
    type: 'fullscreen',
    fullscreen_mode: false
  }
  // the Welcome and Bye object descriptions
  var SayWelcome = {
    type: 'html-keyboard-response',
    stimulus: 'Welcome to this experiment.\n Press the {space bar} to start.',
    choices: [' '],
  }
  var SayBye = {
    type: 'html-keyboard-response',
    stimulus: 'Thank you for your participation.\n Press the {space bar} to end the experiment.',
    choices: [' '],
  }


  // **************************************
  // *      Defining trial-level events      *
  // **************************************

  var SetnBlinkingItems = {
    type: 'call-function',
    func: function() {
      nBlinkingItems =  jsPsych.timelineVariable("OneSequenceLength", true);
      currentBlinkingItem = 0;   // counter that will count the number of blinks
```

```
    currentResponse = 0;         // counter that will count the number of responses
  },
}
var GeneratePositions = {
  type: 'call-function',
  func: PositionFunction
}
var WaitInterTrialDuration = {
  type: 'html-keyboard-response-noerase',
  stimulus: '',
  choices: jsPsych.NO_KEYS,
  trial_duration: InterTrialDuration
}
var WaitPreBlinkDuration = {
  type: 'html-keyboard-response-noerase',
  stimulus: '',
  choices: jsPsych.NO_KEYS,
  trial_duration: PreBlinkDuration
}
var WaitBlinkDuration = {
  type: 'html-keyboard-response-noerase',
  stimulus: '',
  choices: jsPsych.NO_KEYS,
  trial_duration: BlinkDuration
}
var WaitInterBlinkDuration = {
  type: 'html-keyboard-response-noerase',
  stimulus: '',
  choices: jsPsych.NO_KEYS,
  trial_duration: InterBlinkDuration
}
var WaitPostBlinkDuration = {
  type: 'html-keyboard-response-noerase',
  stimulus: '',
  choices: jsPsych.NO_KEYS,
  trial_duration: PostBlinkDuration,
}
var ShowFixation = {
  type: 'html-keyboard-response',
  stimulus: '+',
  choices: jsPsych.NO_KEYS,
  trial_duration: FixationDuration,
  on_start: HideMouse
}
var ShowOneItem = {
  type: 'html-keyboard-response-noerase',
  stimulus: jsPsych.timelineVariable('ItemHTML'),
  choices: jsPsych.NO_KEYS,
  trial_duration: 0
}
var ShowAllItems = {
  timeline: [ ShowOneItem ],
  timeline_variables: ListOfItems
}
var BlinkOneItem = {
  type: 'html-keyboard-response-noerase',
  stimulus: jsPsych.timelineVariable('ItemBLANK'),
  choices: jsPsych.NO_KEYS,
  trial_duration: 0,
}
var BlinkAllItems = {
  timeline: [ BlinkOneItem, WaitBlinkDuration, ShowOneItem, WaitInterBlinkDuration ],
  timeline_variables: ListOfBlinkingItems,
  conditional_function: function() {
    if(currentBlinkingItem++ < nBlinkingItems) {return true;}
    else {return false;}}
```

```
    }

    var StartResponding = {
      type: 'audio-keyboard-response',
      stimulus: RecallSignal,
      choices: jsPsych.NO_KEYS,
      trial_duration: RecallSignalDuration,
      on_start: ShowMouse
    }
    var BlinkThatResponse = {
      type: 'html-button-response',
      stimulus: '',
      button_html: function() {
        var choice = jsPsych.data.get().last(1).values()[0].button_pressed;
        if (choice !== null) {
          return ListOfButtons.concat(ListOfItems[choice].ItemACKNW);}
        else {return "";}
      },
      trial_duration: AcknowledgeDuration,
      choices: function() {return [...Array(ListOfButtons.length+1).keys()]}
    }
    var ReadOneResponse = {
      type: 'html-button-response',
      stimulus: '',
      button_html: ListOfButtons,
      choices: function() {return [...Array(ListOfButtons.length).keys()] },
      trial_duration: 5000
    }
    var ReadAllResponses = {
      timeline: [ ReadOneResponse, BlinkThatResponse ],
      timeline_variables: ListOfBlinkingItems,
      conditional_function: function() {
        if(currentResponse++ < nBlinkingItems) {return true;}
        else {return false;}}
    }
    var GatherResponses = {
      // jsPsych.data collects everything, so needs to filter
      // the odd button presses on the last 2*nBlinkingItems events
      type: 'call-function',
      func: function() {
        var stem = jsPsych.data.get().filter({trial_type: 'html-button-response'}).last(2*
  nBlinkingItems);
        var responses = stem.select('button_pressed').values.filter((a,i)=>i%2===0);
        responses = responses.map(Number);
        var correct   = [...Array(nBlinkingItems).keys()];
        var acc       = responses.every(function(value, index) { return value === correct[index]});
        var rts = stem.select('rt').values.filter((a,i)=>i%2===0)
        console.log(responses, acc, rts);
      }
    }

    var RunOneTrial = {
      timeline: [ SetnBlinkingItems,
                  GeneratePositions,
                  WaitInterTrialDuration,
                  ShowFixation,
                  ShowAllItems,
                  WaitPreBlinkDuration,
                  BlinkAllItems,
                  WaitPostBlinkDuration,
                  StartResponding,
                  ReadAllResponses,
                  GatherResponses // don't forget to save...
              ],
      randomize_order:    false,
      repetitions:        sequenceRepetition
```

```
    }

    var RunAllTrials = {
      timeline:           [ RunOneTrial ],
      timeline_variables: sequenceLengths,
      randomize_order:    sequenceRandomOrder
    }

    jsPsych.init({
      timeline: [ FullScreenOn, SayWelcome, RunAllTrials, FullScreenOff, SayBye ]
    })

  </script>
 </body>
</html>
```

Open practices

The *Open Material* badge was earned because supplementary material(s) are available on the journal's web site.

Citation

Gibeau, R.-M. (2021). The Ccorsi Blocks Task: Variations and coding with jsPsych. *The Quantitative Methods for Psychology*, *17*(3), 299–311. doi:10.20982/tqmp.17.3.p299

Received: 06/08/2021 ∼ Accepted: 06/18/2021