

SOFTWARE METAPAPER

A Working Memory Test Battery: Java-Based Collection of Seven Working Memory Tasks

James M. Stone¹ and John N. Towse¹

¹ Department of Psychology, Lancaster University, UK
j.stone@lancaster.ac.uk
j.towse@lancaster.ac.uk

Working memory is a key construct within cognitive science. It is an important theory in its own right, but the influence of working memory is enriched due to the widespread evidence that measures of its capacity are linked to a variety of functions in wider cognition. To facilitate the active research environment into this topic, we describe seven computer-based tasks that provide estimates of short-term and working memory incorporating both visuospatial and verbal material. The memory span tasks provided are; digit span, matrix span, arrow span, reading span, operation span, rotation span, and symmetry span. These tasks are built to be simple to use, flexible to adapt to the specific needs of the research design, and are open source. All files can be downloaded from the project website <http://www.cognitivetools.uk> and the source code is available via Github.

Keywords: Working Memory; Computerised Cognitive Testing; Cognitive Assessment; Verbal Working Memory; Visuospatial Working Memory, Java; Tatool

Funding Statement: Funded via a PhD Studentship award from the ESRC.

(1) Overview

Introduction

Klatzky (1980) described the working memory system as being a mental workbench using the analogy of a carpenter's workbench. As a carpenter will lay out the tools and materials that they need for the job on their workbench so they are readily available, our mental workbench can hold 'chunks' of information that are required for our current cognitive goals. This workspace has limited capacity and the critical factors that contribute to this capacity 'limit' are open to debate but it is clear that the working memory system is a limited resource and that this limit varies across individuals.

Working memory ability has been shown to correlate reliably with other cognitive abilities such as; fluid intelligence (Conway, Cowan, Bunting, Theriault, & Minkoff, 2002), arithmetic (McLean & Hitch, 1999), the ability to prevent mind wandering during tasks requiring focus (Kane et al., 2007), executive attention (Kane & Engle, 2003), general learning disabilities (Alloway, 2009), and many more. In addition to its prominence within cognitive research there are a wide variety of other disciplines incorporating WM ability in to their research programmes and assessing the impact of this cognitive system on their respective domains of study. Some examples of topics that have seen measures of WM used as a predictor include depression (Arnett et al., 1999), learning computer

languages (Shute, 1991), life event stress (Klein & Boals, 2001), regulating emotion (Kleider, Parrott, & King, 2010), and multitasking (Bühner, König, Pick, & Krumm, 2006; Hambrick, Oswald, Darowski, Rench, & Brou, 2010).

Complex span tasks

Daneman and Carpenter (1980) reported a paradigm that was designed to capture the conceptual requirements of simultaneous processing and memory operations thought to be inherent to working memory functioning. In their reading span task participants were required to read aloud sentences and attempt to remember the last word in each sentence. Administration consisted of three trials at set sizes two through six. The simultaneous processing of information while needing to store information for recall has become an integral part of working memory research.

Complex span tasks follow the paradigm of item storage with concurrent processing of a demanding task in which there are a set number of item storage and cognitive processing events. The form of the to-be-remembered (TBR) items and the processing task can take many forms. Turner and Engle (1989) introduced the operation span task with two versions that differed in the TBR units. The processing part of the operation span task involves presenting the participant with a mathematical operation (e.g. '(6/2) + 2 = 5') to which the participant must assess whether or not the printed answer is correct. In the 'Operations Word'

version each operation was followed by the presentation of a word and these words made up the TBR array. Another version used in their experiment was 'Operations Digit' where the participant was required to recall the numbers that were given as answers to the operations (regardless of whether the operation was true or false). Other variants of a verbal complex span task exist such as the counting span task (Case, Kurland, & Goldberg, 1982) where object counting forms the processing and array totals provide the TBR items, which was designed to be appropriate for a wide developmental population.

A popular thread of research in the working memory literature relates to whether there is a separation of verbal and visuo-spatial domains in the WM system (as popularised by the multi-component model of working memory, Baddeley and Hitch (1974); Baddeley (1986)) or if a domain-general pool of resources might be the driving force behind WM performance. Therefore alongside verbal complex span tasks, a number of visuo-spatial complex span tasks have developed over time. For example, Shah and Miyake (1996) introduced a 'rotation span' task. This combined a processing phase which involved mentally rotating letters and judging whether or not they were regular or mirror images with a storage phase that presented arrows in varying orientations and lengths. The symmetry span task (Kane et al., 2004) uses grid locations in a 5x5 matrix as the storage units while the processing phase requires judgements on the symmetry of a pattern filled in an 8x8 matrix.

Availability of computerised tasks

Given this widespread incorporation of working memory measures in many domains of research, the availability of software to run computerised assessments of aspects of working memory ability is important for the ongoing investigation into the construct itself and the relationship with other functions.

Currently, the choices available to a researcher wishing to use working memory tasks, are (a) to build a software package from the ground up, or (b) to use a commercial product, such as standardised test kits available that relate to working memory such as the automated working memory assessment (Alloway, 2007). The AWMA is sold primarily to the education sector to assess pupils working memory but is also used as a research tool to provide measures of WM e.g. Holmes et al. (2010). One drawback to using such a tool might be cost, administering these packages to hundreds of participants would add a large amount to the cost of an experiment. In terms of experimental design these tools might not be a good fit in that they are administered in a specific way and there is no room for modification. This is a necessary property for a standardised tool so that one can assess scores compared to the normalised scores.

In many cases it seems that when a researcher wants to use a measure of working memory (and indeed for many other 'non-standard' tools for measuring cognition) they produce a version of the task 'in-house'. This allows researchers the flexibility of being able to make design choices that fits their experimental design e.g. custom

length, number of trials at each set size, randomised trial order, control over what data is logged, and many more. The drawback to this paradigm is that there are likely countless versions of each 'non-standardised' working memory task out there that have been developed for a specific research program. A large number of these could have been reused in a way that would have generated a huge saving in resources.

There are some notable examples of computerised working memory tasks that have been published and made available online for other researchers to use. The attention and working memory lab at Georgia Institute of Technology have made available versions of five (at the time of writing) complex span tasks using E-Prime software (Unsworth, Heitz, Schrock, & Engle, 2005) also see Redick et al. (2012). In addition to 'normal' length versions of the tasks, shortened versions are also available (Foster et al., 2014). The availability of these tools is excellent given the extensive research the group have put into validation (Redick et al., 2012). Another freely available set of computerised tasks for assessment of working memory have been produced using Matlab (Lewandowsky, Oberauer, Yang, & Ecker, 2010). This battery consists of four tasks that the researchers selected to be representative of the various facets of the WM construct and therefore provide a reliable and valid measure of WM ability.

These tasks are presented in the form of scripts that can be executed in their respective programs (E-Prime/Matlab). Therefore with some knowledge of the programming in these frameworks one could modify them to change elements of the tasks. However, the scripts can only be executed on computers with the E-Prime/Matlab installed and this involves expensive license fees. Many Universities/Departments pay these license fees and therefore may have computers with the programs installed for researchers to use, or one may have to incorporate the costs of the software into any grant proposal. This issue can be a wider concern if your research involves going out and collecting data outside of the lab. For example, in developing a working memory training experiment as part of a PhD thesis we conducted an experiment that involved pupils at a number of schools carrying out working memory tasks in a group setting. To do this we needed to use the IT facilities that the school had. It would not have been practical to use currently available systems to collect such data.

A further tool which provides researchers with the means to create computerised cognitive tasks is the Psychology Experiment Building Language (PEBL; Mueller & Piper, 2014). PEBL provides a framework for creating tasks but also includes a battery of commonly used tasks with the install, some of which are working memory tasks. PEBL is an open source project. Of the seven tasks I present here, there are versions of five of them in the PEBL battery (digit span, reading span, operation span, matrix span, and symmetry span). There are subtle differences in the implementation of these tasks between the versions in the PEBL battery and the versions presented here. The PEBL is an excellent project and has evolved into an immensely useful tool for researchers. There are many tasks available in

the PEBL battery beyond just the WM tasks we provide here. With regards to the WM tasks I think using PEBL or the software described here is a choice for each researcher to make based on the computers you are testing on (PEBL requires an install) and the ease with which the task can be configured to the specification desired.

Rationale for software described in this paper

There are a number of computerised tools available to a researcher interested in measuring WM. However, each one has its own limitations; either a rigid administration or lack of flexibility in platform it can be used on. For these reasons, we present our suite of computerised WM tasks. Built using Tootool, a Java-based framework (von Bastian, Locher, & Ruffin, 2013) and entirely open source. As the tasks are based on Java they are easily run on various operating systems providing the Java Runtime Environment (JRE) is installed. The JRE is often pre-installed on machines given the widespread use of Java and therefore no install is usually required to run the application described in this paper. Should a JRE not already be present on the target machine it is freely and easily accessible online.

We describe here a set of tasks commonly used tasks to assess short-term/working memory, a number of complex span tasks described above as well as simple span counterparts for both the verbal and visuo-spatial domain. Each task is independent and so there is no requirement to administer all of them as a package. Instead the pool of tasks is made available so that anyone can select the most appropriate based on your research needs. Measuring working memory ability is best achieved through administering a number of tasks and forming a composite score from them (Kane, Hambrick, & Conway, 2005; Lane et al., 2004). However, sometimes this is not possible. Perhaps time spent with the participant is very limited or there are already a number of computerised tasks and fatigue is a concern. In these instances one can select the task(s) that most suit your needs based on your research goals.

Throughout the rest of this paper we will outline the tasks that are currently available, where to get them, how to use them, what we think they offer that is not already readily available, and some commentary on the ongoing nature of the project.

Tasks

Verbal WM tasks

There are three verbal span tasks currently available, each a slight variation on the others.

Operation Span. The operation span is perhaps the most commonly used verbal complex span task. **Fig. 1** shows a schematic view of how a trial in the operation span task is executed. Most complex span tasks can be broken down into a repeated cycle of memory and processing components. The current version of an operation span task consists of presenting the participant with an integer that needs to be stored and recalled at the end of the trial in its correct serial position. For every storage element (integer to remember) there is a processing phase immediately succeeding it. The processing phase presents the participant with a mathematical operation such as

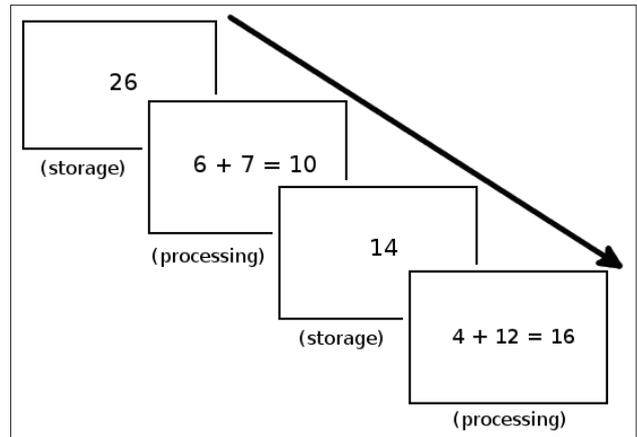


Figure 1: Illustration of the operation span task.

' $6 + 7 = 10$ ' and the participant must indicate if they think the given answer is correct or not. The digits and operations are randomly generated each trial. The minimum and maximum digit can be set as an option in the module file, the default is digits between 10 and 99. For each operation there is a .5 probability that the operation will be correct and the type of operation (multiplication, division, addition, subtraction) used has a .25 probability for each, this should provide a variety of types of operation requiring correct and incorrect responses.

Processing-Storage order. It is worth mentioning that the 'traditional' method of administering complex span tasks such as the operation span task involves using a processing-storage order of phases rather than storage-processing as we have used. This method is rather curious as the processing task serves the purpose of adding to the cognitive demands of storage by requiring processing of a task while trying to store stimuli. Therefore with a processing-storage order of presentation it seems that the first processing phase of a trial has a different effect to subsequent processing phases of the trial. In the first instance the participant is not holding any TBR stimuli and thus this processing phase is not being carried out while task specific items are being stored in WM (for data consistent with the notion that the first episode has different requirements see Towse, Hitch, and Hutton (1998)). That is not to say that the first processing element does not add something to the cognitive requirements above a simple span measure but that the effect it has should be considered differently to the subsequent processing phases. In this operation span task the similarity is clearly very high as the constituent parts of the processing tasks are numbers, as are the TBR items. There is also then the property of the participant being shown the final TBR item immediately followed by the recall phase with no effective retention interval. Therefore this item has not been subject to any additional processing decay. A span size two trial illustrates the point most strikingly. With a processing-storage order the two TBR items bookend a processing operation. The recall screen will appear and the second TBR item will have only just been presented. If the participant can recall the first item that was one processing phase ago then they can almost always be successful at these trials. It would seem that having this order could be

considered a methodological choice that exacerbates the recency effect.

Reading Span. The reading span task (Fig. 2) differed from the operation span task only in the processing element. Rather than having to verify a mathematical operation, the participants were presented with a sentence; their task was to decide if it made sense or not. A note of caution when using this task is that the sentences themselves are defined in a stimuli file and any sentence is only used once. Therefore if you require more trials than the provided stimuli file can accommodate you would need to use their own sentences even in the case that the provided stimuli file contains enough items.

Digit Span. The digit span task is the memory span equivalent to the operation/reading span tasks. It is operationally the same as these tasks but with no processing phase therefore simply a stream of digits that must be remembered in serial position.

Spatial WM tasks

Symmetry Span. The symmetry span task is a spatial complex span task. Participants are required to remember grid (4x4) locations presented to them in the correct serial order. Fig. 3 shows a schematic representation of this task. As is shown, participants are given a processing operation to complete after each TBR grid is presented.

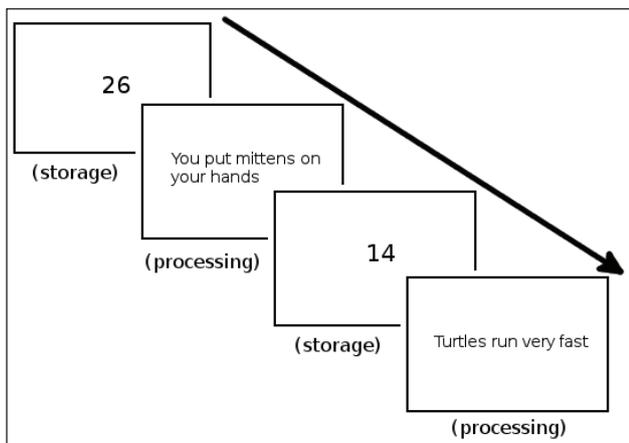


Figure 2: illustration of the reading span task.

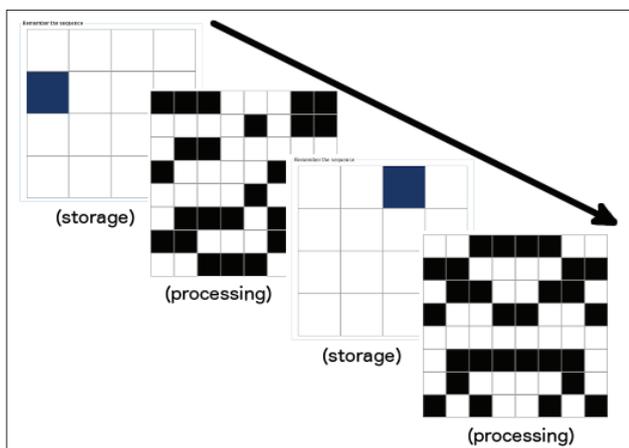


Figure 3: Illustration of the symmetry span task.

This processing element requires them to make a judgement of whether the presented pattern is symmetrical along the vertical axis or not using the left/right arrow keys (8x8 grid used for presenting patterns).

After the appropriate number of storage-processing elements have run for a trial the recall phase begins. Responses are recorded by presenting participants with the 4x4 grid and allowing them to click the boxes in the order they recall seeing them. When a box is selected it turns blue so participants can keep track of their responses.

The size of the grids (4x4 for storage and 8x8 for processing) is customisable in the module file as well as how large they appear on screen.

Matrix Span. The matrix span task is the memory span equivalent of the symmetry span task. The procedure is the same as described for symmetry span except for the removal of the processing element.

Rotation Span. Fig. 4 shows a schematic representation of a rotation span trial showing the storage and processing parts of the task. The to-be-remembered (TBR) stimuli in the rotation span task are images of arrows that are differentiated in two characteristics. Any one arrow can differ in its length (long or short), or it can differ in its angle of rotation (0°, 45°, 90°, 135°, 180°, 225°, 270°, or 315°). Therefore the storage phase of this task is to remember the arrows presented in their correct serial position.

The processing operation in this complex span task presents participants with a letter (F, G, or R) that may be standard or a mirror image. It may also be rotated at one of the 45 degree rotations. The participant must mentally rotate the image so that they can make a judgement on whether the letter is a normal or mirror representation using the left/right keys.

The recall screen presented the 16 possible arrows in a 2x8 grid where the top row of arrows were all the short arrows and the bottom row were all the long arrows. Participants used the mouse to select the arrows they remembered seeing in the correct order.

Arrow Span. The arrow span task is the memory span equivalent to the rotation span task. Therefore the processing phase is dropped such that the arrow span task is simply about remembering the arrows in correct serial position.

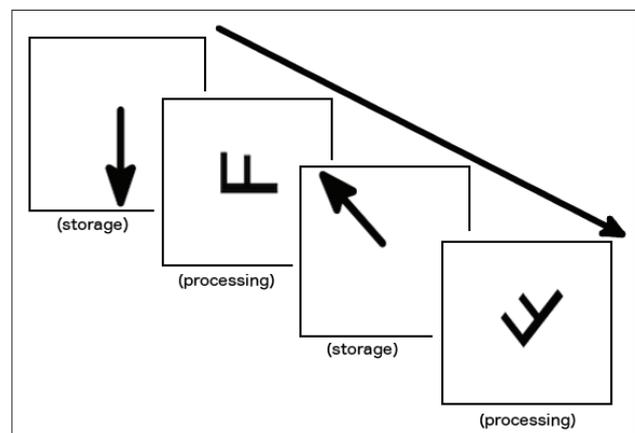


Figure 4: Illustration of the rotation span task.

Implementation and architecture

The software is based on the Java programming language and therefore will be compatible with any computer with a Java runtime installed (currently tested and running perfectly on the latest JRE which is version 8 update 31). Therefore, the programs are accessible on multiple operating system platforms, including windows, OSX, and linux. The framework for the tasks is provided by Tatoon (von Bastian et al., 2013). Tatoon is an open source platform/library that provides much of the functionality required for creating computerised psychological tasks. Using Tatoon still involves programming a set of executables that run the type of task desired but Tatoon provides a number of functions to make this easier as well as providing a neat presentation of the program and storage of user information and results.

Tatoon and the working memory executables described here are also open source. This means that users are encouraged to engage with the source code and modify it to their own specification if the functionality required is not provided by default. For example, Tatoon applications and modules can be extended to interact with a web server for remote administration of tasks. This is a built in Tatoon feature and instructions on adding this functionality are included in the Tatoon download (<http://www.tatoon.ch/download.htm>). It is hoped that as users add functionality these modifications can be added to the project. This is an important feature going forward but for the rest of this paper we discuss the tasks as they are currently configured and specified.

The project website can be found at the following URL: <http://www.cognitivetools.uk>. The website hosts the application as well as the supporting files all of which will be discussed below. Using these tasks does not require any interaction with the source code. But we do provide the source code in a github repository (DOI: 10.5281/zenodo.14825) and would welcome anyone to fork the project and help improve and add to the project. This code needs to be added to a maven project with Tatoon version 1.3.2 installed.

Instructions on using the tasks

There are two components needed to run any of these tasks and an optional third component that can help process the resultant data from using this package. The first component is the application itself. The application we provide is an executable jar and this jar file launches just like any other executable if an up to date Java runtime is running on the operating system. The jar contains all of the Java dependencies as well as the Tatoon libraries. In addition to these components it contains the executables we have written to run the tasks described above. The app contains all the source code but will do nothing without the use of module files.

Tatoon is designed to use xml files that specify (amongst other things) for the application which executables to run in what order. What this means for a user is that after launching the application first time a module file for each task needs to be loaded. I provide a default module file for each task. Inside the module file a number of values that will affect the behaviour of the program can be changed.

Task flexibility

As we have alluded to already everything is technically flexible as the source code is provided and it is encouraged that users modify/improve where they see fit to achieve the results required.

The current suite of tasks have been designed and arranged to permit flexibility in their usage. Depending on hypotheses being tested and practical restrictions such as duration constraints the deployment of a span task may need to be altered from the default settings. By building the executables in a way that certain task behaviours can be easily altered by values in the xml the user is not restricted in their deployment. It is important to remind users that a rigid administration of span measures may be required for comparisons across studies.

Each task has a dedicated webpage, and one of the sections on these pages outlines what variables can be customised using values in the module files. If we take the symmetry span task as an example, **Table 1** outlines the

Variable	Potential Values	Result
SpanTwoTrials–spanNineTrials	Any Integer	Informs the program how many trials at each span size to run.
RandomisedTrials	0/1	If set to 0 then the trials will be given in ascending span size order. If set to 1 then the order of trials is randomised.
GridDimension	Any Integer (but be sensible...)	The value (n) set here will be the size of the matrix presented to participants. The default is 5 which produces a 5x5 grid.
ScalingFactor	1–100	Sets the size of the grid shown to participants. The program will work out how much space there is available for the grid presentation. The scaling factor is then applied to determine the resultant size of the display. For example if set to 50 then the grid size is half the maximum.
MinSquares	Integer < half the total number of squares	In conjunction with the 'MaxSquares' value, sets the minimum and maximum number of squares to fill when creating a pattern for symmetry judgement.
MaxSquares	Integer < half the total number of squares	See above.

Table 1: Properties that can be modified with simple value changes in the module file for the Symmetry Span task.

information provided in the customisation section of the symmetry span task page. Some of these variables are common to many tasks such as the number of trials to run at each span size, and whether or not you want these trials to run in order or randomised. If you elect to run them in order (non-randomised) then the program will run the specified amount of trials at each span size starting at the lowest through to the highest, so if you have set three at each size then it will run three trials at span size two followed by three trials at span size three and so forth. However, some researchers may wish to randomise the trial order so that the participant isn't aware on any trial how many TBR items they are going to be presented with (e.g. Engle, Cantor, & Carullo, 1992).

As well as these attributes that are common across the tasks there are other attributes that can be altered that only apply to a subsection of the tasks or are unique to a task. An example of this is the 'gridDimension' variable which controls the size of the matrix used to deliver grid locations to participants and is present in the matrix and symmetry span tasks.

Loading a module file

Fig. 5 outlines the basic procedure for preparing an instance of one of the tasks for a participant. To load a module file one must first make a user profile in the Tatoon application. Tatoon was created primarily to help researchers who conduct experiments that involves training (multi-session experiments) on the psychological tasks

created and therefore much of its functionality is designed with that in mind. Therefore we must load a user profile to begin serving the tasks to our participants. The user information provided here is saved in every output file created from tasks administered to that user profile. Therefore it is possible to create user accounts for participants and load up specific modules for them. This is not necessarily required though, as each task will ask the experimenter to input a 5-digit participant code at the start of a trial. This allows the experimenter to create one user account, load the necessary modules for that testing session, and then supply different participant codes on each administration. For more complex designs the experimenter may wish to create different users for different participants, while for more simple designs the experimenter can rely on the supplied participant codes.

Once a user has been created we are taken to the users module page, when we load a module file we add to the list of modules a user can access. To load a module click the 'Add' button in the bottom-left corner and then select 'Module from local file'. Navigate to the directory that stores the module file and select it. If there are no errors in the module file then a new module will appear in the 'My Modules' pane (Fig. 5 box 3). Once the module is loaded, changes to the module file are not reflected in the execution of that module. For example, if it is decided to modify the module file so as to administer more trials then the updated module file needs to be reloaded.

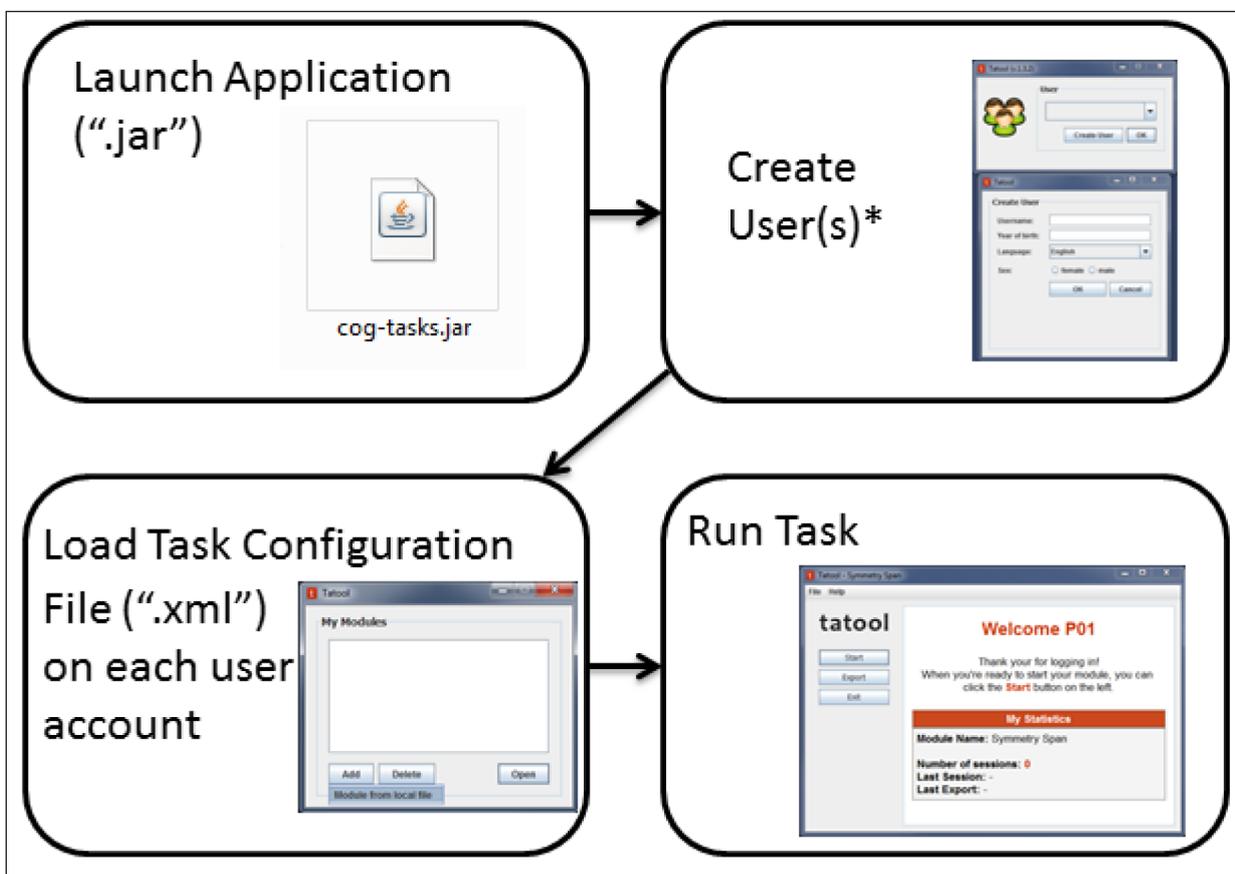


Figure 5: Flow diagram displaying the basic steps needed to launch a task.

Extracting the data

Each time a participant completes a trial the data is saved automatically. This data can be extracted in CSV format using the 'Export' button. The resultant CSV file will contain information on all trials that the user account has completed. As hinted above there are two likely methods that a researcher will have deployed these tasks; either a user account per participant with the necessary modules loaded, or single account with each different participant being identified by providing a 5-digit participant code at the start of each administration. We provide two R (R Core Team, 2014) scripts per task, one for each method. The r-scripts are organised with the name format 'x_span_process.r' and 'x_span_process_singleUser.r'

Single account method

This simpler method will result in just one CSV file being exported at the end of the data collection phase of your experiment. Download the appropriate R-script from the website (making sure to use the script with '_singleUser' suffix). Simply open the script and change "datafile.csv" to point to your data file; a) if the working directory of R is set to the location of the data file then simply put the file name here, or b) alternatively supply an absolute file path or a file path relative to the current working directory. Then execute the script and a data frame will be in the R workspace called 'x.span.data' which summarises each participants performance.

Multiple account method

This method will result in an output CSV file per participant. Take all these files from the respective task and put them in the same directory with no other files. The directory that holds the data files needs to be set as the working directory in the R environment and then the respective R script can be run; either by using the source() function, or opening the script within R and highlighting and running the code. We provide links to useful resources on the website for those who are unfamiliar with R. Once again after executing the script the result will be a new data frame in the R workspace which summarises each participants performance on that task.

General comments on data processing

The scripts are written to extract general summary information about performance on the respective tasks. Examples of measures calculated:

- Number of trials successfully recalled at each span size.
- Accuracy of processing phase responses (complex span measures).
- Median response time for processing phase responses (complex span measures).
- Various 'scores' to reflect overall performance.

Each task webpage includes a data section which outlines what all the variables produced by the script represent. The obtained data frame after executing the provided R script can be analysed within R or extracted to be analysed in an alternate data analysis program.

A final note about the R scripts is that they have been produced to work with the tasks as they are. This means that if you change the source code to modify a task then it is also possible you may need to change the R script to analyse the output. Modifications might include the introduction of new variables or the renaming of existing variables which would require an updated R script to process. This issue relates to actual source code modification of the executables, the R scripts are flexible to the various alterations that can be made by the user within the module file.

Quality control

The software described in this paper has been used in two large scale experiments by the authors as well as two smaller experiments as part of the lead authors PhD research.

(2) Availability

Operating system

OSX/Windows/Linux. Any OS with an up to date JRE installed.

Dependencies

Using the application only requires a JRE installed. To engage with the source code a number of maven dependencies are required (tatool version 1.3.2 and the dependencies that is reliant). Using the provided pom.xml file in the repository and an IDE with maven functionality should automatically download the required packages.

Software location

The compiled ready-to-use software is hosted at the project website, the downloads page can be found at <http://www.cognitivetools.uk/cognition/downloads>. The project website hosts the main application as well as the additional default module files and R analysis scripts. The source code for the project is hosted in a GitHub repository.

Archive and code repository

GitHub

Name

cog-tasks: Working Memory Test Battery

Persistent identifier

10.5281/zenodo.15488

URI

<https://github.com/j-stone/cog-tasks.git>

Project homepage

<http://www.cognitivetools.uk/>

Licence

GNU GPL v3.0

Publisher

James M Stone

Date published

08/02/2015

(3) Reuse potential and summary

Tasks that assess working memory performance are being used widely across many domains of research. Often research groups need to create their own version of described tasks due to a lack of freely available resources that are flexible in their implementation and easy to deploy on a variety of platforms. In this paper we have introduced open source simple/complex span tasks that are built using Tatoon, a Java-based platform, and will provide researchers with measures of both verbal and visuospatial working memory.

These tasks can be downloaded and used without the need for any code wrangling immediately by using the default settings. Additionally, they can be altered via the built in customisable options by simply changing values in the module files. And finally, as the source code for both Tatoon (the framework) and our executables are open source any modifications that one wishes to make can be made. Additionally, data processing scripts are provided which will process the resultant data using R.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

We would like to thank Claudia von Bastian and Andre Locher for their work developing Tatoon and for their helpful suggestions on an earlier draft of this manuscript.

References

- Alloway, T P** 2007 *Automated working memory assessment: Manual*. Pearson.
- Alloway, T P** 2009 Working memory, but not iq, predicts subsequent learning in children with learning difficulties. *European Journal of Psychological Assessment*, 25(2): 92–98. DOI: <http://dx.doi.org/10.1027/1015-5759.25.2.92>
- Arnett, P, Higginson, C, Voss, W, Wright, B, Bender, W, Wurst, J and Tippin, J** 1999 Depressed mood in multiple sclerosis: Relationship to capacity-demanding memory and attentional functioning. *Neuropsychology*, 13(3): 434–446. DOI: <http://dx.doi.org/10.1037/0894-4105.13.3.434>
- Baddeley, A D** 1986 *Working memory*. London: Oxford University Press.
- Baddeley, A D and Hitch, G** 1974 Working memory. In G. H. Bower (Ed.), *The psychology of learning and motivation*, pp. 47–89. New York: Academic Press. DOI: [http://dx.doi.org/10.1016/S0079-7421\(08\)60452-1](http://dx.doi.org/10.1016/S0079-7421(08)60452-1)
- Bühner, M, König, C, Pick, M and Krumm, S** 2006 Working memory dimensions as differential predictors of the speed and error aspect of multitasking performance. *Human Performance*, 19(3): 253–275. DOI: http://dx.doi.org/10.1207/s15327043hup1903_4
- Case, R, Kurland, D M and Goldberg, J** 1982 Operational efficiency and the growth of short-term memory span. *Journal of experimental child psychology*, 33(3): 386–404. DOI: [http://dx.doi.org/10.1016/0022-0965\(82\)90054-6](http://dx.doi.org/10.1016/0022-0965(82)90054-6)
- Conway, A R, Cowan, N, Bunting, M F, Theriault, D J and Minko, S R** 2002 A latent variable analysis of working memory capacity, short-term memory capacity, processing speed, and general fluid intelligence. *Intelligence*, 30(2): 163–183. DOI: [http://dx.doi.org/10.1016/S0160-2896\(01\)00096-4](http://dx.doi.org/10.1016/S0160-2896(01)00096-4)
- Daneman, M and Carpenter, P** 1980 Individual differences in working memory and reading. *Journal of verbal learning and verbal behavior*, 19(4): 450–466. DOI: [http://dx.doi.org/10.1016/S0022-5371\(80\)90312-6](http://dx.doi.org/10.1016/S0022-5371(80)90312-6)
- Engle, R W, Cantor, J and Carullo, J J** 1992 Individual differences in working memory and comprehension: a test of four hypotheses. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 18(5): 972–992. DOI: <http://dx.doi.org/10.1037/0278-7393.18.5.972>
- Foster, J L, Shipstead, Z, Harrison, T L, Hicks, K L, Redick, T S and Engle, R W** 2014 Shortened complex span tasks can reliably measure working memory capacity. *Memory & Cognition*, pp. 1–11. DOI: <http://dx.doi.org/10.3758/s13421-014-0461-7>
- Hambrick, D, Oswald, F, Darowski, E, Rench, T and Brou, R** 2010 Predictors of multitasking performance in a synthetic work paradigm. *Applied Cognitive Psychology*, 24(8): 1149–1167. DOI: <http://dx.doi.org/10.1002/acp.1624>
- Holmes, J, Gathercole, S, Dunning, D, Hilton, K, Elliott, J, et al.** 2010 Working memory deficits can be overcome: Impacts of training and medication on working memory in children with ADHD. *Applied Cognitive Psychology*, 24(6): 827–836. DOI: <http://dx.doi.org/10.1002/acp.1589>
- Kane, M J, Brown, L H, McVay, J C, Silvia, P J, Myin-Germeys, I and Kwapil, T R** 2007 For whom the mind wanders, and when an experience-sampling study of working memory and executive control in daily life. *Psychological science*, 18(7): 614–621. DOI: <http://dx.doi.org/10.1111/j.1467-9280.2007.01948.x>
- Kane, M J and Engle, R W** 2003 Working-memory capacity and the control of attention: the contributions of goal neglect, response competition, and task set to stroop interference. *Journal of experimental psychology: General*, 132(1): 47–70. DOI: <http://dx.doi.org/10.1037/0096-3445.132.1.47>
- Kane, M J, Hambrick, D and Conway, A** 2005 Working memory capacity and fluid intelligence are strongly related constructs: comment on Ackerman, Beier, and Boyle (2005). *Psychological Bulletin*, 131(1): 66–71. DOI: <http://dx.doi.org/10.1037/0033-2909.131.1.66>
- Kane, M J, Hambrick, D Z, Tuholski, S W, Wilhelm, O, Payne, T W and Engle, R W** 2004 The generality of working memory capacity: a latent-variable approach to verbal and visuospatial memory span and reasoning. *Journal of Experimental Psychology: General*, 133(2): 189–217. DOI: <http://dx.doi.org/10.1037/0096-3445.133.2.189>
- Klatzky, R L** 1980 *Human memory: structures and processes*. San Francisco: Freeman.
- Kleider, H, Parrott, D and King, T** 2010 Shooting behaviour: How working memory and negative emotionality influence police officer shoot decisions. *Applied Cognitive Psychology*, 24(5): 707–717. DOI: <http://dx.doi.org/10.1002/acp.1580>

- Klein, K** and **Boals, A** 2001 The relationship of life event stress and working memory capacity. *Applied Cognitive Psychology*, 15(5): 565–579. DOI: <http://dx.doi.org/10.1002/acp.727>
- Lewandowsky, S, Oberauer, K, Yang, L** and **Ecker, U K** 2010 A working memory test battery for matlab. *Behavior Research Methods*, 42(2): 571–585. DOI: <http://dx.doi.org/10.3758/BRM.42.2.571>
- McLean, J F** and **Hitch, G J** 1999 Working memory impairments in children with specific arithmetic learning difficulties. *Journal of Experimental Child Psychology*, 74(3): 240–260. DOI: <http://dx.doi.org/10.1006/jecp.1999.2516>
- Mueller, S T** and **Piper, B J** 2014 The psychology experiment building language (PEBL) and PEBL test battery. *Journal of neuroscience methods*, 222: 250–259. DOI: <http://dx.doi.org/10.1016/j.jneumeth.2013.10.024>
- R Core Team** 2014 R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from <http://www.R-project.org>
- Redick, T S, Broadway, J M, Meier, M E, Kuriakose, P S, Unsworth, N, Kane, M J** and **Engle, R W** 2012 Measuring working memory capacity with automated complex span tasks. *European Journal of Psychological Assessment*, 28(3): 164–171. DOI: <http://dx.doi.org/10.1027/1015-5759/a000123>
- Shah, P** and **Miyake, A** 1996 The separability of working memory resources for spatial thinking and language processing: an individual differences approach. *Journal of Experimental Psychology: General*, 125(1): 4–27. DOI: <http://dx.doi.org/10.1037/0096-3445.125.1.4>
- Shute, V** 1991 Who is likely to acquire programming skills? *Journal of Educational Computing Research*, 7(1): 1–24. DOI: <http://dx.doi.org/10.2190/VQJD-T1YD-5WVB-RYPJ>
- Towse, J N, Hitch, G J** and **Hutton, U** 1998 A re-evaluation of working memory capacity in children. *Journal of memory and language*, 39(2): 195–217. DOI: <http://dx.doi.org/10.1006/jmla.1998.2574>
- Turner, M** and **Engle, R** 1989 Is working memory capacity task dependent? *Journal of Memory and Language*, 28(2): 127–154. DOI: [http://dx.doi.org/10.1016/0749-596X\(89\)90040-5](http://dx.doi.org/10.1016/0749-596X(89)90040-5)
- Unsworth, N, Heitz, R, Schrock, J** and **Engle, R** 2005 An automated version of the operation span task. *Behavior Research Methods*, 37(3): 498–505. DOI: <http://dx.doi.org/10.3758/BF03192720>
- von Bastian, C C, Locher, A** and **Ruffin, M** 2013 Tatoon: A java-based open-source programming framework for psychological studies. *Behavior research methods*, 45(1): 108–115. DOI: <http://dx.doi.org/10.3758/s13428-012-0224-y>

How to cite this article: Stone, J M and Towse, J N 2015 A Working Memory Test Battery: Java-Based Collection of Seven Working Memory Tasks. *Journal of Open Research Software* 3:e5, DOI: <http://dx.doi.org/10.5334/jors.br>

Published: 05 June 2015

Copyright: © 2015 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License (CC-BY 3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/3.0/>.