We here present DiViDu, a research software, which we developed for a dual task experiment with an integrated divided visual field paradigm. The dual task experiment consists of a lexical decision task with different semantic verb categories, and a complex tapping task with either the left or right hand. The software allows researchers the replication of our experiment, as well as the reconfiguration for further dual task experiments with alternative tapping tasks and stimuli (e.g., word classes, semantic classes, task languages) as well as adjusted experimental settings. Furthermore, with only slight modifications to the source code, researchers can implement various new experimental setups including a large variety of language tasks (e.g., silent and aloud reading, naming, verbal fluency) and non-verbal tasks. The software DiViDu is implemented using the .NET framework and is available under the Apache 2 License on GitHub (https://github.com/dividu/dividu).

Keywords: neuropsychology; behavioural experiments; lateralisation; dual task paradigm; divided visual field paradigm

Funding statement: The publication of the paper was supported by the Open Access Publication Fund of the University of Konstanz.
In the following, we describe the experimental setup, the implementation, the requirements as well as the full range of adaptional potential.

Implementation and architecture

Experimental Setup

We developed DiViDu for a dual task experiment, where participants perform either a lexical decision task only (single task) or simultaneously with a complex one-handed tapping task (dual task) (cf., [11] for an overview). The participants perform the tapping task either with their right or left hand, while responding to the lexical decision task with the other hand. Furthermore, all stimuli are presented in the right visual field (RVF) and the left visual field (LVF) (divided visual field paradigm) (cf., Figure 1). The word stimuli contain of hand-related (H) [e.g., clap, write], as well as foot-related (F) action verbs [e.g., run]. We further subdivided the category of hand-related actions words into a group of unimanual (H1) and a group of bimanual (H2) hand-related actions words. The verbs in category H1 describe actions normally performed with only one hand [e.g., write], while the verbs in category H2 describe actions normally performed with both hands simultaneously [e.g., clap].

Every lexical decision sequence consists of the presentation of a fixation cross (800 ms) and a stimulus (180 ms), a response period (3000 ms maximum) and an interstimulus interval (1200 ms) (cf., Figure 4).

The complex one-handed tapping task consists of six steps (2-3-4-5-2-5) with the index (2), middle (3), ring (4) and little (5) finger of the right or left hand.

The participants enter the lexical decision response (word/pseudoword) as well as the complex tapping task sequence on a regular, but rotated, keyboard (cf., Figure 1). In our experiment, one-half of the participants (group R) performed the tapping task with the right and responded to the lexical decision task with the left hand. We recorded F2 (word) and F1 (pseudoword) key inputs for the lexical decision response, as well as F9, F10, F11 and F12 for the tapping input. Furthermore, we recorded the reaction time for the lexical decision responses as well as the sequence of the tapping keys and the intertap intervals, reflecting the period between key down of one of the tapping keys and key down of the next tapping key. We subdivided the experiment into different parts, separated by breaks. The participants determine the length of the breaks. Moreover, the participants initiate the start of the entire experiment as well as the continuation of the experiment after a break. In our experiment, we defined breaks after every 80 stimuli presentation. At the beginning of the experiment, as well as when changing task condition (from a dual task condition to single task condition or the reverse), participants see an instruction screen for 3000 ms (cf., Figure 2).

In addition to the main experiment, our participants performed a training task (cf., Figure 3), a dual task setting with a tapping task and a non-verbal task. Therefore, the software displays a yellow or red bar instead of a word or pseudoword. We instructed our participants to respond according to the colour but not the position of the bar. During the training task, the response 'yellow' is entered with the word key, while the response 'red' is entered with the pseudoword key. We implemented the training task to accustom the participants to a dual task, as well as a visual field experimental setting.

We implemented DiViDu using the most straightforward programming technology to ensure that scientists with
limited programming experience can understand and change the source code. We use the programming language Visual Basic in the Visual Studio 2017 framework. Visual Studio has a very convenient debugging interface, which allows less experienced programmers to experience a direct connection between the current state of graphical user interface and the position in the source code.

Our software has a global state. The main loop iterates through a trial list until the end of the list is reached. The three data sources, the keystrokes for the tapping input and the lexical decision input, as well the information from the trial list, are written into a joint log file.

In addition to state altering keypress events, we use a stopwatch control to record the duration between the keystrokes (reaction time for the lexical decisions and intertap intervals for the tapping). Furthermore, we use a stopwatch (timer) to advance the state in between the sequence. The flow diagram of our software DiViDu is shown in Figure 4.

**Trial list**
We defined the stimuli, as well as the presentation settings and the task condition for each stimulus in four different trial lists (cf., Table 1). Our original German trial lists are provided via GitHub. Table 1 shows exemplarily a section of the trial list in English. The trial list includes the stimuli (1) and the visual field setting (2) [LVF/RVF]. Furthermore, it specifies the stimulus type (3) [pseudoword (TRUE)/word (FALSE)], the task condition (4) [dual task condition (TRUE)/single task condition (FALSE)], the semantic word categories (5), as well as the subcategories (6).

**Output file**
The software provides a separate CSV output file for each participant. The output table (cf., Table 2) includes the date and time of the experiment conduct and a comment (2) which signalises the instruction period at the beginning of a new task condition. Thus, row 2 reflects no lexical decision results, but the key input (tapping sequence and intertap intervals) during the instruction period. The following columns reflect the stimuli information including the stimulus type (3), the visual field setting (4), the task condition (5), the table includes the stimuli (6), the word category (7) as well as the subcategory (7). Finally, it contains the correctness of the response (correct/incorrect/timeOver) (10) and the reaction time (11). Thus, ‘timeOver’ reflects a missing response after a defined response period, in our experimental setup 3000 ms.

Moreover, a column TappingKeys (12) display the key inputs for the tapping sequence. In addition, the column intertap interval (13) shows the intertap intervals, reflecting the tapping frequency.

**Quality control**
We provide a test trial list in English, which allows developers to perform a manual test with overall 20 stimuli. The test trial lists will automatically be installed with the software, the setup covers all possible states of our software so that the user can quickly ensure the integrity and stability of the software. The only requirement for this

---

**Table 1:** Exemplary trial list for the lexical decision task.

<table>
<thead>
<tr>
<th>Stimulus (1)</th>
<th>VisualField (2)</th>
<th>Pseudoword (3)</th>
<th>Tapping (4)</th>
<th>Wordcategory (5)</th>
<th>SubCategory (6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>run</td>
<td>LVF</td>
<td>FALSE</td>
<td>TRUE</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>lun</td>
<td>LVF</td>
<td>TRUE</td>
<td>TRUE</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>plap</td>
<td>RVF</td>
<td>TRUE</td>
<td>TRUE</td>
<td>H</td>
<td>H2</td>
</tr>
<tr>
<td>clap</td>
<td>RVF</td>
<td>FALSE</td>
<td>TRUE</td>
<td>H</td>
<td>H2</td>
</tr>
<tr>
<td>run</td>
<td>LVF</td>
<td>FALSE</td>
<td>TRUE</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>lun</td>
<td>RVF</td>
<td>TRUE</td>
<td>FALSE</td>
<td>H</td>
<td>H2</td>
</tr>
<tr>
<td>plap</td>
<td>LVF</td>
<td>TRUE</td>
<td>FALSE</td>
<td>H</td>
<td>H2</td>
</tr>
<tr>
<td>clap</td>
<td>RVF</td>
<td>FALSE</td>
<td>FALSE</td>
<td>H</td>
<td>H2</td>
</tr>
<tr>
<td>write</td>
<td>LVF</td>
<td>FALSE</td>
<td>FALSE</td>
<td>H</td>
<td>H1</td>
</tr>
<tr>
<td>grite</td>
<td>LVF</td>
<td>TRUE</td>
<td>FALSE</td>
<td>H</td>
<td>H1</td>
</tr>
</tbody>
</table>
test is a break-distance setting smaller than the stimuli number (e.g., 5), to reach and test the break state. After every modification of the source code, the user should carefully check the following physical parameters of the experimental setup: (1) Correct size of the stimuli on the screen with respect to the screen resolution and the physical dimensions of the screen and (2) Correct timing of the experimental trials and the recording of the time periods and (e.g., with a high-speed camera).

Additional system requirements
To collect valid data for scientific purposes, users should ensure the following requirements and provide the following equipment. To download and use the software, users need a Windows PC and a screen with a minimal size of 15-inch. After the start of the software, users have to specify their screen size in the settings. The software is preconfigured a distance of 60 cm between the eyes of the participants and the screen. The software is preconfigured a distance of 60 cm between eyes and screen. See [2] for recommendations regarding the position of stimuli presentation (visual angle) and presentation period in divided visual field experiments. Additionally, an input device (number pad or regular keyboard) is needed. Again, the delay of the keyboard should be considered for the precision of the reaction time for the lexical decision input as well as the intertap intervals. Furthermore, a low-stimulus environment (i.e., appropriate seating, quiet, well-tempered room) as well as controlled light condition, preferably dimmed room with a single light source on the ground, is essential for reliable participants performance.

Dependencies
Compiled for target framework .NET 4.6.1 (however other target frameworks are selectable).

Software location
Code repository
Name: GitHub
Identifier: https://github.com/dividu/dividu
Licence: Apache License, 2.0
Date published: Latest release 05/10/2017

Language
English

(3) Reuse potential
Behavioural experiments regarding the interaction of the motor and the language system include various language and motor task. The language tasks comprise lexical decision [3], [8], reading [5], naming [12], verbal fluency [6] and verbal memory tasks [7]. Similarly, the complexity of the motor task highly various between the experiments including only one [6] or both hands [7]. Furthermore, the task and stimuli language, the experimental setup as well as the used software differs between the experiments [4–8], [12]. Open Source software allows researchers to reproduce results, as well as to modify only one or a few aspects of an experimental setting to produce comparable results. In the following, we will describe the reuse potential of our experiment by adapting, the trial lists, the settings as well as the source code.

The modifications of the trial lists allows users to modify the stimuli type and language.

Settings
Furthermore, the adaption of the settings (cf., Figure 5) at the beginning of the experiment, allows users to modify the software to their specific needs regarding timing of the presentation sequence, the number and location of the input keys and the labels of the instruction screens. The settings from the last usage remain stored as defaults for the next usage. Thus, it is easy to perform a sequence of experiments with identical settings.

The user can specify the input keys, for the tapping task as well as for the language task, in number and location. Thus, the software allows a large variety of tapping tasks, ranging from a simple one-finger tapping to a complex tapping sequences with up to five fingers. As DiViDu

### Table 2: Output file.

<table>
<thead>
<tr>
<th>Comment</th>
<th>PseudoWord</th>
<th>VisualField</th>
<th>Task</th>
<th>Stimulus</th>
<th>WordCategory</th>
<th>SubCategory</th>
<th>Response</th>
<th>ReactionTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>TapTest</td>
<td>FALSE</td>
<td>LVF</td>
<td>TRUE</td>
<td>run</td>
<td>Foot</td>
<td>F</td>
<td>Correct</td>
<td>0</td>
</tr>
<tr>
<td>FALSE</td>
<td>LVF</td>
<td>TRUE</td>
<td>run</td>
<td>Foot</td>
<td>F</td>
<td>TimeOver</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TRUE</td>
<td>RVF</td>
<td>TRUE</td>
<td>plap</td>
<td>Hand</td>
<td>H2</td>
<td>Correct</td>
<td>1040</td>
<td></td>
</tr>
<tr>
<td>FALSE</td>
<td>LVF</td>
<td>TRUE</td>
<td>clap</td>
<td>Hand</td>
<td>H2</td>
<td>Correct</td>
<td>733</td>
<td></td>
</tr>
<tr>
<td>TapTest</td>
<td>TRUE</td>
<td>RVF</td>
<td>FALSE</td>
<td>lun</td>
<td>Foot</td>
<td>F</td>
<td>Correct</td>
<td>671</td>
</tr>
<tr>
<td>TRUE</td>
<td>RVF</td>
<td>FALSE</td>
<td>lun</td>
<td>Foot</td>
<td>F</td>
<td>Correct</td>
<td>760</td>
<td></td>
</tr>
<tr>
<td>TRUE</td>
<td>LVF</td>
<td>FALSE</td>
<td>plap</td>
<td>Hand</td>
<td>H2</td>
<td>Incorrect</td>
<td>564</td>
<td></td>
</tr>
<tr>
<td>FALSE</td>
<td>RVF</td>
<td>FALSE</td>
<td>clap</td>
<td>Hand</td>
<td>H2</td>
<td>Incorrect</td>
<td>1320</td>
<td></td>
</tr>
</tbody>
</table>

CsvHelper (v2.16.3) from NuGet developed by Josh Close (Apache 2 and Microsoft Public Licence).
records the tapping sequence, the experimental setup allows not only the analysis of the tapping frequency, but also the analysis of errors in the tapping order.

For the language tasks, binary decisions, as well as decisions with more than two response keys can be performed. If the experimental design includes two groups that perform the tasks with different hands, the user can define the tapping hand at the beginning of the experiment.

To easily adapt the experiments for participants with different languages and different tasks, users can specify the labels for the start, break and final instruction as well as for the task instruction.

Moreover, the user can define the duration of the individual states in the presentation sequence (task instruction, fixation cross-presentation, stimulus presentation, response screen as well as the interstimulus interval).

The instructor can define the time between the start of the software and the beginning of the experiment, as well as the break duration. Alternatively, the instructor can choose an extended period and let the participant start the experiment, as well as initiate the continuation of the experiment after a break of desired length, by pressing the ESC key. Furthermore, the number of stimuli to be performed between two breaks can be defined accordingly to the entire number of stimuli and the complexity of the dual task setting.

**Adaption of the Source Code**

The software structure allows additional experimental setups for a large variety of behavioural experiments. In the following, we present some possible adaption for future research.

**Language tasks**

Instead of a here performed lexical decision task, the software structure provides the opportunity for a large variety of language tasks in dual task designs. Tasks include reading tasks either silently or aloud. A divided visual field paradigm, due to the stimuli location as well as the short presentation period, limits the stimuli to words. Presenting the stimuli in a central location (excluding a divided visual field paradigm) allows reading tasks with phrases or sentences. Furthermore, semantic as well as phonological decision task can be implemented by adapting the input keys and the stimuli. Moreover, by implementing pictures instead of verbal stimuli, a naming task can be implemented. A more challenging adaption includes speech recording or audio output for verbal memory or speech comprehension tasks.

**Motor tasks**

We here describe a dual task experiment with a complex one-handed tapping task. As mentioned above, multiple different one-hand tapping tasks can be implemented by adapting the settings at the beginning of the experiment. Furthermore, the adaption of the source code, as well as the usage of alternative input devices, allow a huge variety of different motor tasks. These include motor tasks with other body parts (i.e., feet). Language tasks that do not require key input (e.g., reading, verbal fluency, verbal memory tasks), allow motor task input with both hands or feet.

**Nonverbal tasks**

The implementation of pictures instead of word stimuli, can either be used for naming tasks [13], or non-verbal tasks. Thus, the experiment is not limited to investigating the interaction of the language and the motor system, but can also be used for other research fields.

**Divided Visual Field Paradigm**

Our software uses a divided visual field paradigm and thus presents all stimuli in the LVF and RVF. Minor adaption of the software allows the presentation of stimuli in both visual field or a central presentation (excluding divided visual field paradigm).

We here use a simple fixation cross to ensure the central fixation prior to the stimulus presentation in the LVF or RVF, and instruct the participants to fixate the fixation cross. The presentation of a letter or digit, which participants report afterwards, increases the central fixation rate, but can also interact with the following task [2].

In conclusion, our software DiViDu offers a variety of possibilities for neuropsychological research experiments. By adapting the settings as well as the source code, it allows users with and without programming skills to adapt the software to their specific needs. Furthermore, as freely available software, it helps researchers to replicate...
the experiments of other research groups as well as to investigate other fields, but comparable experimental setups.

Support
Users are encouraged to report issues on GitHub and make pull request.

Competing Interests
The authors have no competing interests to declare.

References


Submitted: 12 October 2017 Accepted: 12 March 2018 Published: 06 April 2018

Copyright: © 2018 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See http://creativecommons.org/licenses/by/4.0/.

Journal of Open Research Software is a peer-reviewed open access journal published by Ubiquity Press

OPEN ACCESS