Experimenter instructions

Contents

[**1. General information** 2](#_Toc1589333)

[1.1. Temporal bisection task 2](#_Toc1589334)

[**2. Experimental protocol** 3](#_Toc1589335)

[DAY 1 INSTRUCTIONS 3](#_Toc1589336)

[DAY 2 INSTRUCTIONS 4](#_Toc1589337)

[**2.1. Experiment starts...** 4](#_Toc1589338)

[**3. Run the task on a new computer** 8](#_Toc1589339)

[Folder structure 8](#_Toc1589340)

[Running the script 8](#_Toc1589341)

[Controller buttons 9](#_Toc1589342)

[Key 5 pulses 9](#_Toc1589343)

[Timing accuracy 10](#_Toc1589344)

[Stimulus size 10](#_Toc1589345)

[3.1. The script: step-by-step details 11](#_Toc1589346)

# **1. General information**

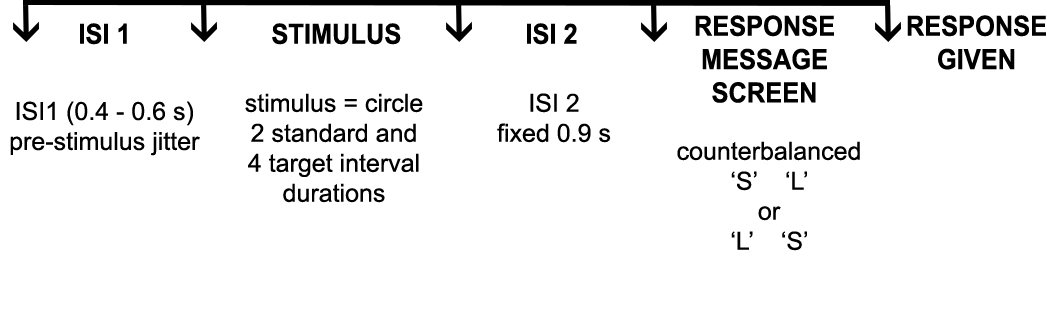
This task measures interval timing (perceived duration and temporal precision) for visual intervals in the sub-second range.

* There will be two experimental sessions – **DAY 1** (pre-surgery) and **DAY 2** (surgery).

## 1.1. Temporal bisection task

Each session begins with the participant learning, through a succession of trials, two different standard (anchor) intervals: a short interval (0.5 s) and a long interval (1.1 s) (*Training phase*). The visual stimuli appear as a circle (2-4 °VF) at the centre of the screen for each interval. The *Testing phase* follows immediately. The participant observes the same circles presented at different target intervals (0.65, 0.75, 0.85, 0.95 s) and judges whether each is closer to the previously-learnt **S**hort or **L**ong standard interval. They use a controller to respond when the Response message screen (letters ***S*** and ***L*** in counterbalanced order) appears.

* Single trial:



* DAY 1 differs in that there are only four blocks of trials in the *Testing phase* and the participant receives the detailed task instructions and experiment description in preparation for DAY 2.
* The *Training phase* includes 20 trials (10 repetitions of each standard). The *Testing phase* on DAY 1 and DAY 2 includes four and six blocks of 50 trials, respectively. Each block includes 2 repetitions of the anchor intervals (i.e. 4 trials) at the beginning (in the order S, L, S, L); the participant is instructed to use these to refresh their memories of the short and long anchors. The remaining 46 trials are randomised target intervals and six more anchor reminders (three each). In the entire task, there are 60 repetitions of each target interval and 30 repetitions of each anchor in the *Testing phase*. The duration of pilot experiments (*Training* + *Testing phase*) with healthy individuals was ~20 min.

# **2. Experimental protocol**

The patients will be familiarized with the experimental task on **DAY 1** prior to the surgery. On **DAY 2**, they will complete a longer version of the task with fewer instructions.

1. Please pilot the experiment and send us the data files so that we can double-check that everything is saved ok.
2. The script times the experiment and the duration of individual blocks. Piloting with patients will help us determine the suitable length of the study.

### DAY 1 INSTRUCTIONS

**The experimenter reads out/ summarizes the instructions in bold:**

**This study investigates time perception. You will complete the task using the index fingers of your left and right hand using these two buttons [**the experimenter shows the two shoulder controller buttons**].**

**There will be a very short training phase followed by a longer testing phase. In the training phase you will be repeatedly presented with two durations of a white circle on the computer screen – one will be “short” and the other will be “long”. After each circle is presented, you should wait until you see two letters appear on the screen (S for “short” and L for “long”). One of the letters will be to the right and the other to the left of the center, corresponding with the right and left buttons on the controller. Press the button based on the circle you just saw.**

**Your job in the training phase is to estimate and memorize these durations as best you can. In the testing phase, you will again see a series of circles of varying duration. Your job will be to judge as best you can whether each circle was closer in duration to the previously-learned short or long durations.**

**Throughout the task, try your best to focus on the circles in the centre of the screen. If your mind ever drifts away, don’t worry – just bring your attention back to the screen. Some people try to use various strategies such as humming or counting to help them estimate the durations. This interferes with your performance and can often cause you to do worse so please don’t do that – just focus on the circles and judge whether they’re closer in duration to the short or long duration. Please note that we’re not interested in how quickly you respond so you need not try to respond super quickly – just pay attention and try your best to judge correctly. Some of the circle durations might be quite difficult to judge – don’t worry about this – just try your best. If you’re unsure, just try to make your best guess – never respond randomly. Please note that many of the circles are “short” and many are “long”. If you find yourself pressing short or long most of the time, you might be doing the task incorrectly – keep this in mind as you complete the task.**

**The experimenter will further explain and examine whether …**

* Does the patient understand that the experiment will be the same on DAY 2, with fewer instructions?
* Does the patient understand the *Training* and *Testing phases*?
* Does the patient understand the task and how to respond and the importance of not responding too prematurely?

Once started, the experimenter and patient may interact, particularly through a succession of the first few trials of each experimental phase to ensure that the task is being completed competently and the patient understands what is expected of them.

### DAY 2 INSTRUCTIONS

It can be counter-productive to go through too detailed instructions at the start of the experiment on DAY 2. The experimenter should aim for a concise refresher - below. Instead, more time should be spent going through the instructions that will appear on the screen at various stages of the experiment. The experimenter reads out (or summarizes) the following text and makes sure that the patient understood.

**The experimenter reads out/ summarizes the instructions in bold:**

**Your task is identical to the one you completed on DAY 1. You will again start with a training phase in which you will learn two circle durations: one is short and the other long. Afterwards, in the testing phase, you will see a series of circles of varying durations and judge whether each one is closer to the previously-learned short or long durations. You will use the same controller buttons as before when the letters S and L appear on the screen.**

**The experimenter will further explain and examine whether …**

* Does the patient understand the *Training* and *Testing phases*?
* Does the patient understand the task and how to respond and the importance of not responding too prematurely?

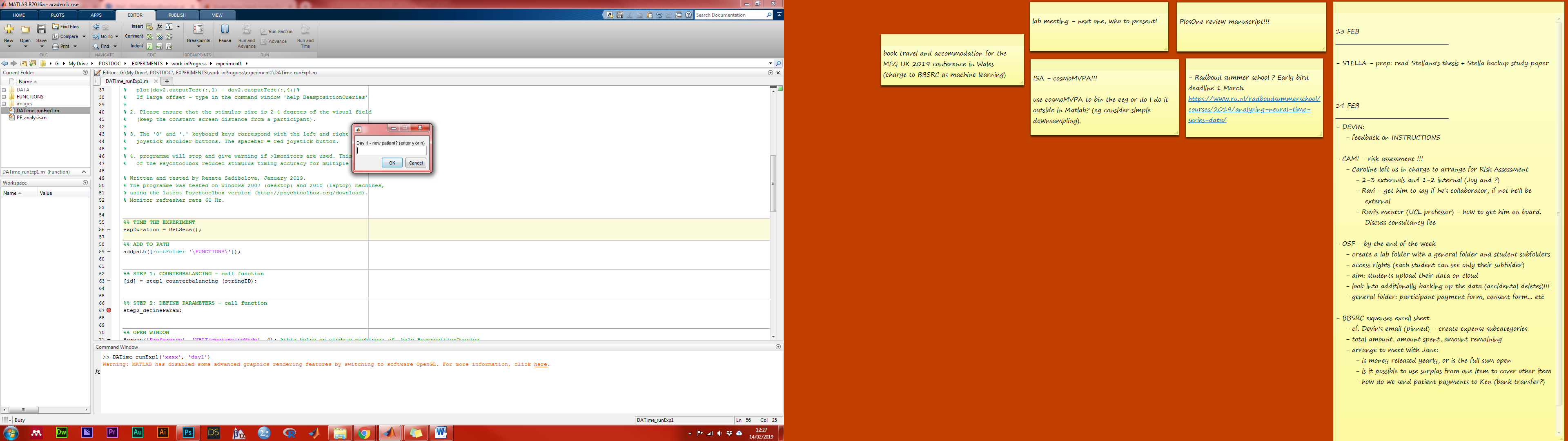
## **2.1. Experiment starts...**

To run the task each time, open in Matlab *current folder* the folder with the function ***DATime\_runExp1.m*** so the function is visible.

The function can be run in three modes (‘day1’, ‘day2’, and ‘testScript’). Please use ‘testScript’ for all piloting and script-testing (refer to section 3 for more details).

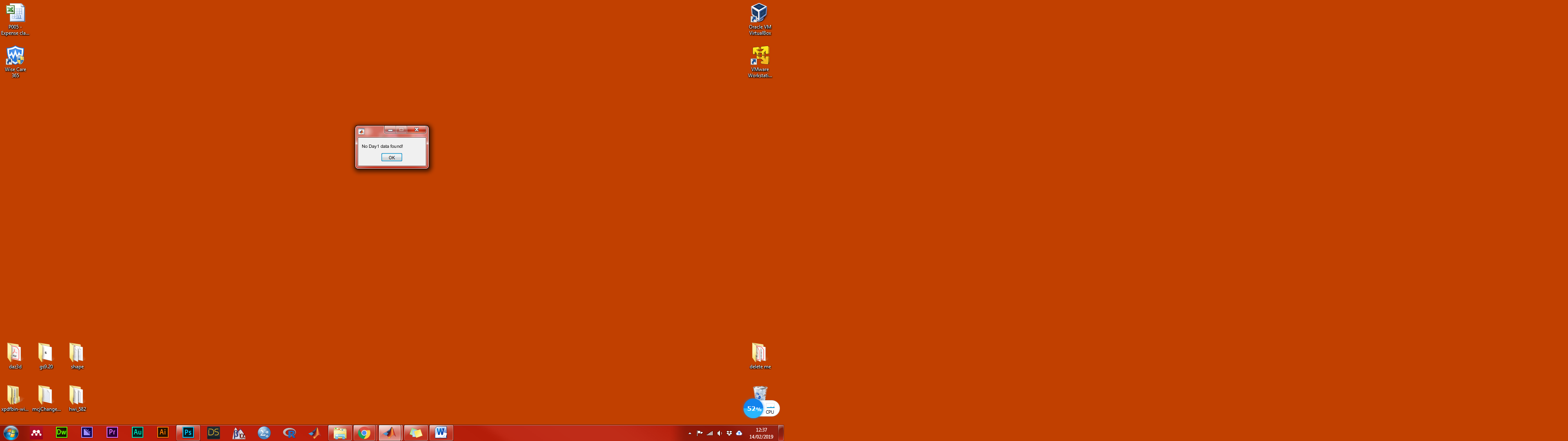
The ‘day1’ and ‘day2’ options must be used **only for actual patients**. The script determines the patient’s testing order number on DAY1 from the count of existing folders (and creates new patient subfolders in the DAY1 and DAY2 folders). This means that you can enter **any** patient ID label that is used in the hospital. The patient’s testing order number, as determined from the folder count, will be stored and used for counterbalancing in the study.

**OPTION 1 (testing the patient on DAY 1)**

Type in the Matlab command window the function name followed by an actual patient ID and the ‘day1’ selection: **DATime\_runExp1 (‘subjID’,’day1’)**

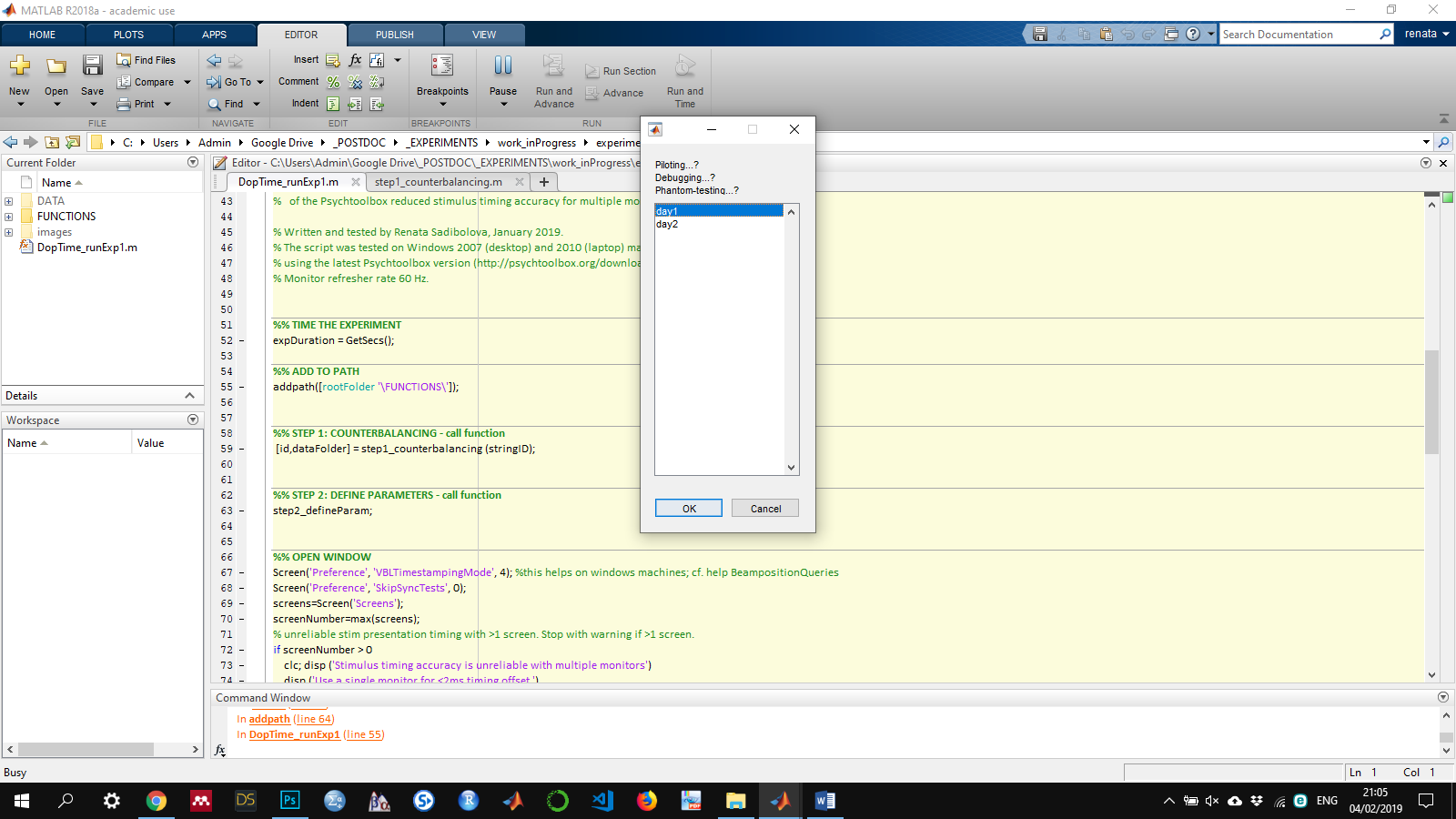
* You’ll be asked to confirm the new patient before the new patient folder is created (their order number, used for counterbalancing, will be determined from a count of existing patient folders).
* If you type in *n*, the script will switch to test mode (OPTION 3 below).
* The data will be saved in the patient’s subfolder in the DAY 1 folder (cf. Section 3 for folder structure information).

**OPTION 2 (testing the patient on DAY 2)**

Type in the Matlab command window the function name followed by an actual patient ID and the ‘day2’ selection: **DATime\_runExp1 (‘subjID’,’day2’)**

* If the patient with this ID was tested on DAY 1, the script will automatically continue with no GUI. Otherwise, if there is no day1 data for the patient, a warning will pop up and the script will continue in the test mode (OPTION 3 below).
* The data will be saved in the patient’s subfolder in the DAY 2 folder (cf. Section 3 for folder structure information).

**OPTION 3 (any piloting and programme-testing)**

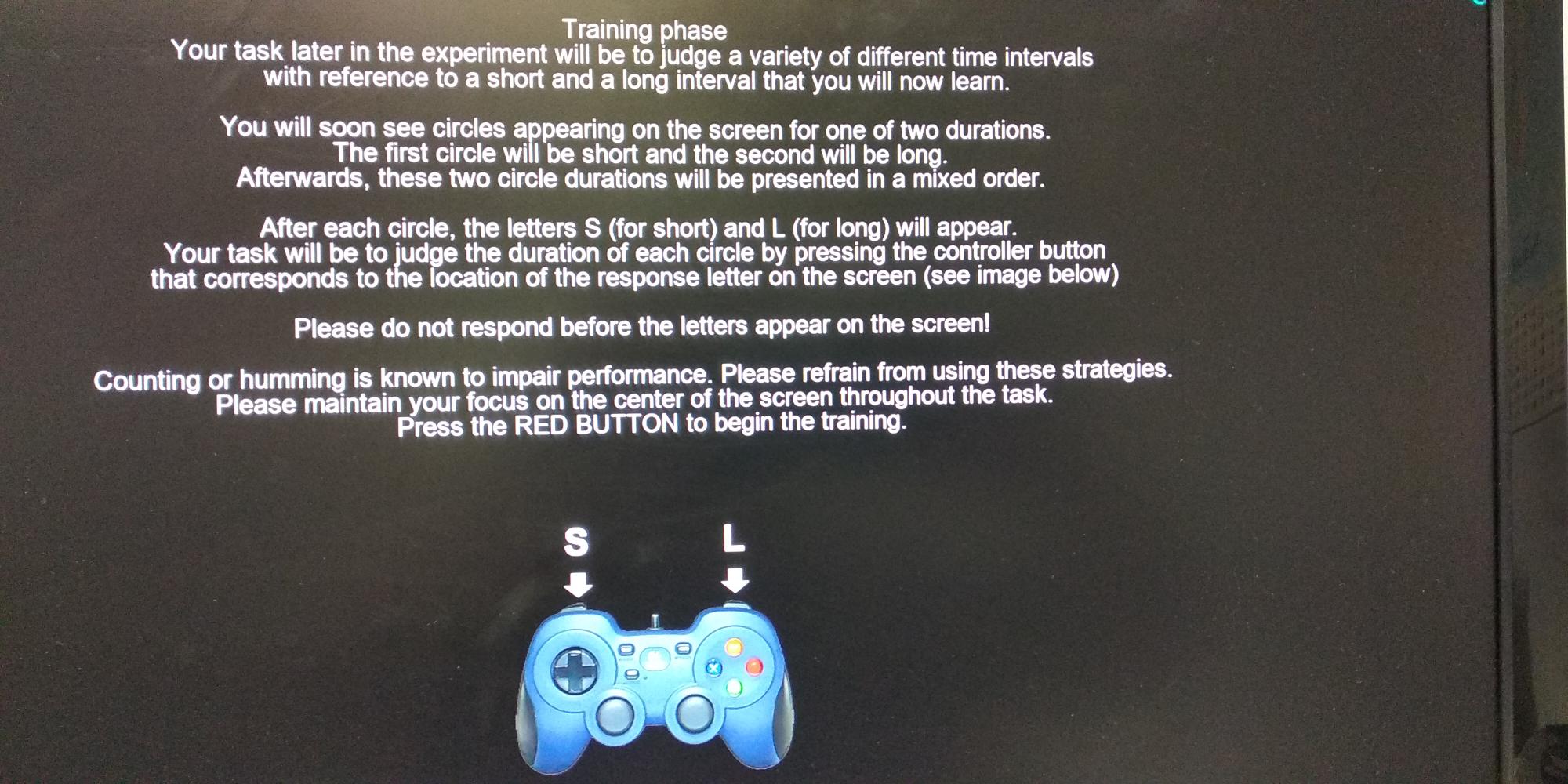
Type the function name in the Matlab command window followed by a string ID of your choice and the ‘testScript’ selection; e.g.:

**DATime\_runExp1 (‘patient\_pilot1’,’testScript’)**

or

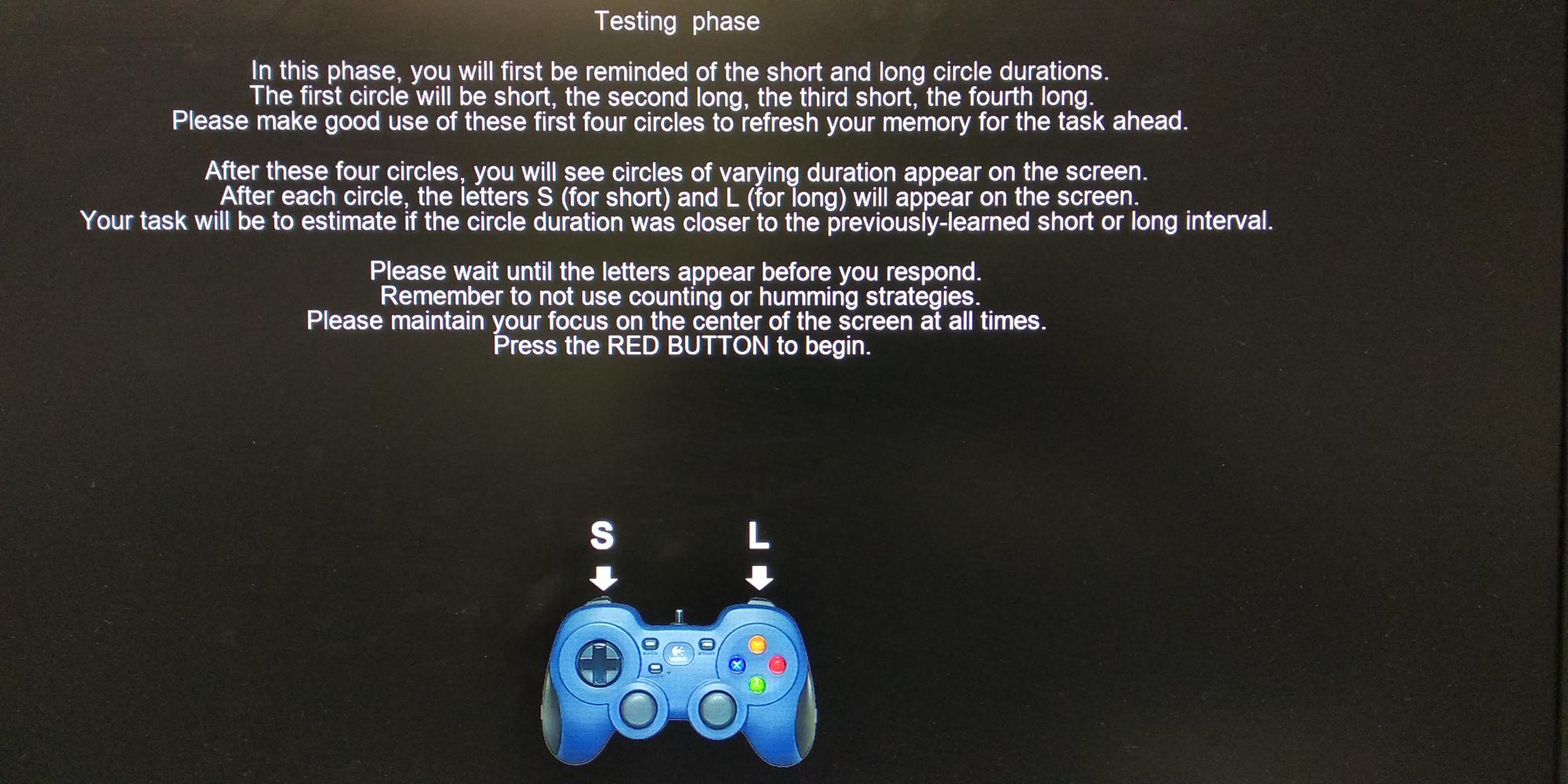
**DATime\_runExp1 (‘XX\_healthyControl’,’testScript’)**

* You’ll be then asked to select which day to pilot:
* The final output file will be stored in newly created folder (its name = string ID of your choice). It will be saved in the TestScript folder (cf. Section 3 for folder structure information).
* The programme will then open the instructions screen for the *Training phase* and it will wait for the patient’s response (red controller button) indicating that the Instructions were read. Please read them out loud and make sure they are understood.



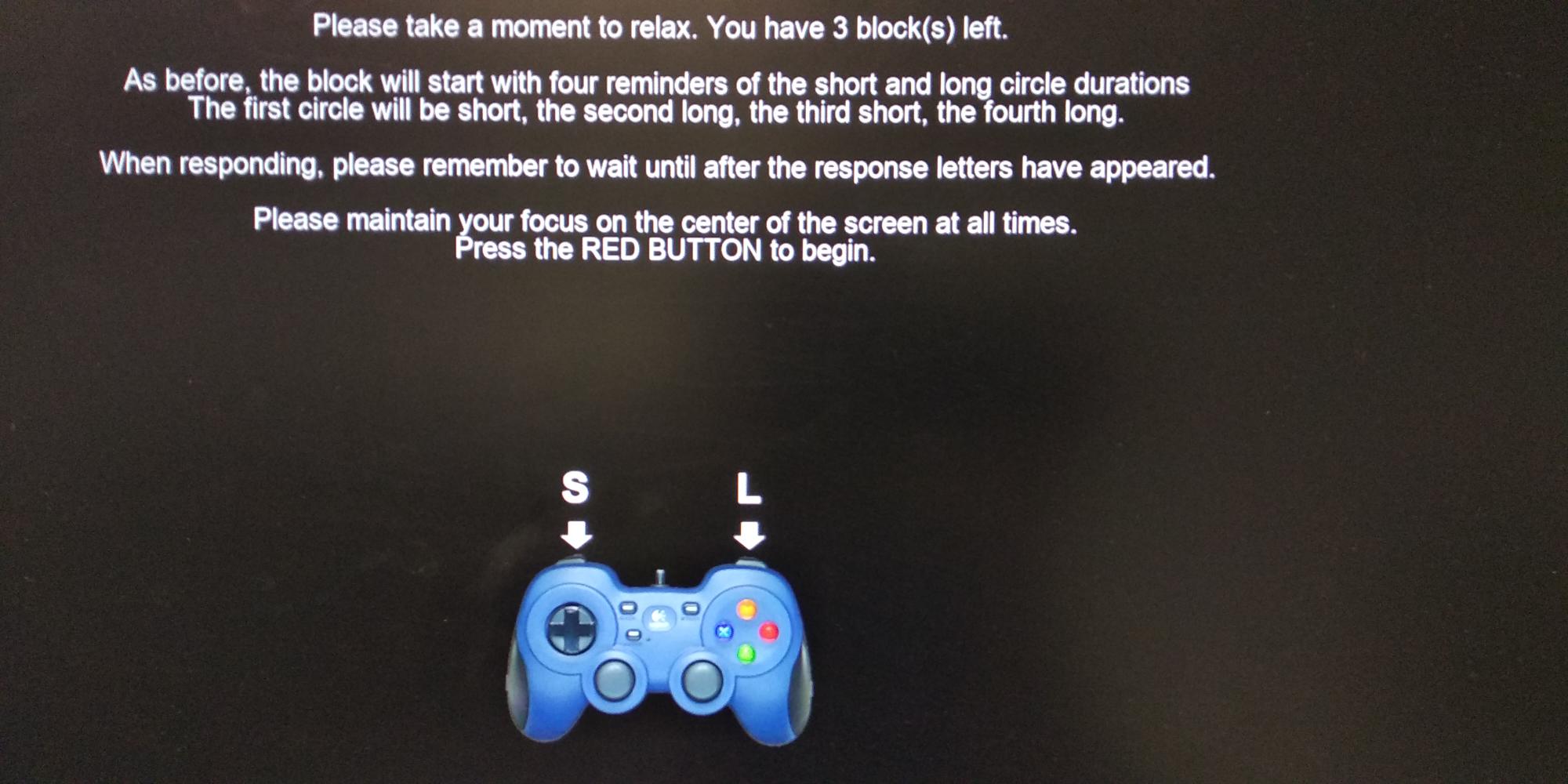
Additional tasks for the experimenter at this stage:

* + Ensure that the patient can see the circles and response letters on the screen.
  + On DAY 1, the experimenter should ask participant to report their responses for the first couple of trials in an interactive manner, and verify how the patient performs and understand the task.
* The patient first completes the *Training phase* of twenty trials. Next, the programme opens the instructions screen for the *Testing phase*. As before, please read them out loud, encourage questions and make sure the patient understands them well.



Additional tasks for the experimenter at this stage:

* + Inform the patient that the testing is organized in six blocks of trials to allow for short breaks to rest one’s eyes, etc. The experimenter should discourage long breaks that would significantly increase the duration of the experiment.
  + The experimenter may remind participants not to respond prematurely. The time window between the stimulus offset and the appearance of the response message screen is an important decision-making period which would otherwise be contaminated by premature button-pressing.
  + The experimenter explains that some trials in the *Testing phase* will be more difficult than others. It’s normal to feel unsure about some of the responses. Encourage the patient to simply try their best and to try to stay focused.
* The programme shows a shorter instructions page with the remaining number of blocks after each block has finished, and waits for the red button press to continue:



* Once the programme has run its course and the task is completed, the window closes and Matlab appears. The duration of the experiment is displayed in the Matlab command window. The individual block durations are saved in the data file.

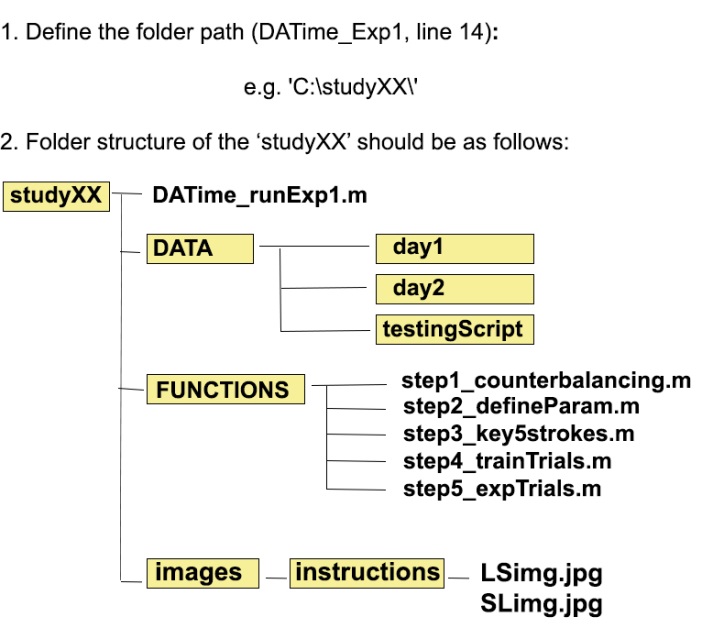
End of the experiment:

1. Ask participants if they used counting or humming strategies.
2. Ask for feedback on the task and how would they assess their overall performance.
3. Ask them about premature responding – how successful they feel they were in avoiding it.
4. Verify the data is saved and back it up if needed.

# **3. Run the task on a new computer**

### Folder structure

* Start by creating a folder on your computer where the script and data will be saved.
* Define the path to this folder in ***DATime\_runExp1.m*** task script, in line 14.
* The task script ***DATime\_runExp1.m*** runs with a predefined folder structure (please see image below). Please ensure the content of the folder you created corresponds with this structure (or download it created for you from the shared Google drive folder).

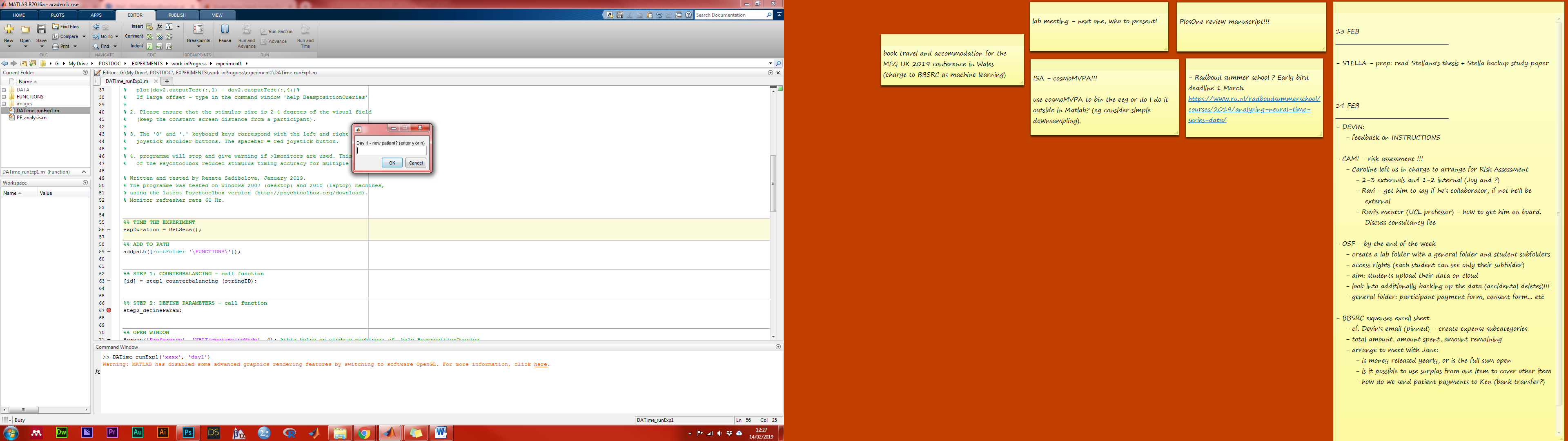


### Running the script

* To run the task each time, open in Matlab *current folder* the folder with the function ***DATime\_runExp1.m*** so the function is visible.

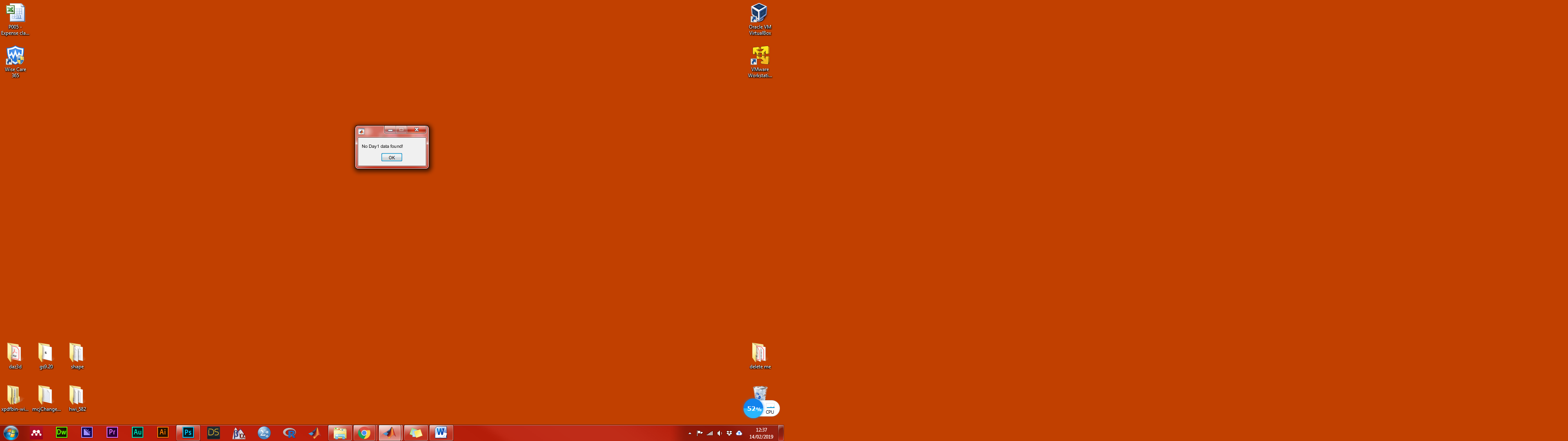
The function can be run in three modes (‘day1’, ‘day2’, and ‘testScript’). Please use ‘testScript’ for piloting and script-testing. The ‘day1’ and ‘day2’ options must be used **only for actual patients**. The script determines the order number of each new patient on DAY1 from the count of existing folders (and creates new patient subfolders in the DAY1 and DAY2 folders). The order number is stored and used for counterbalancing.

**OPTION 1 (testing the patient on DAY 1)**

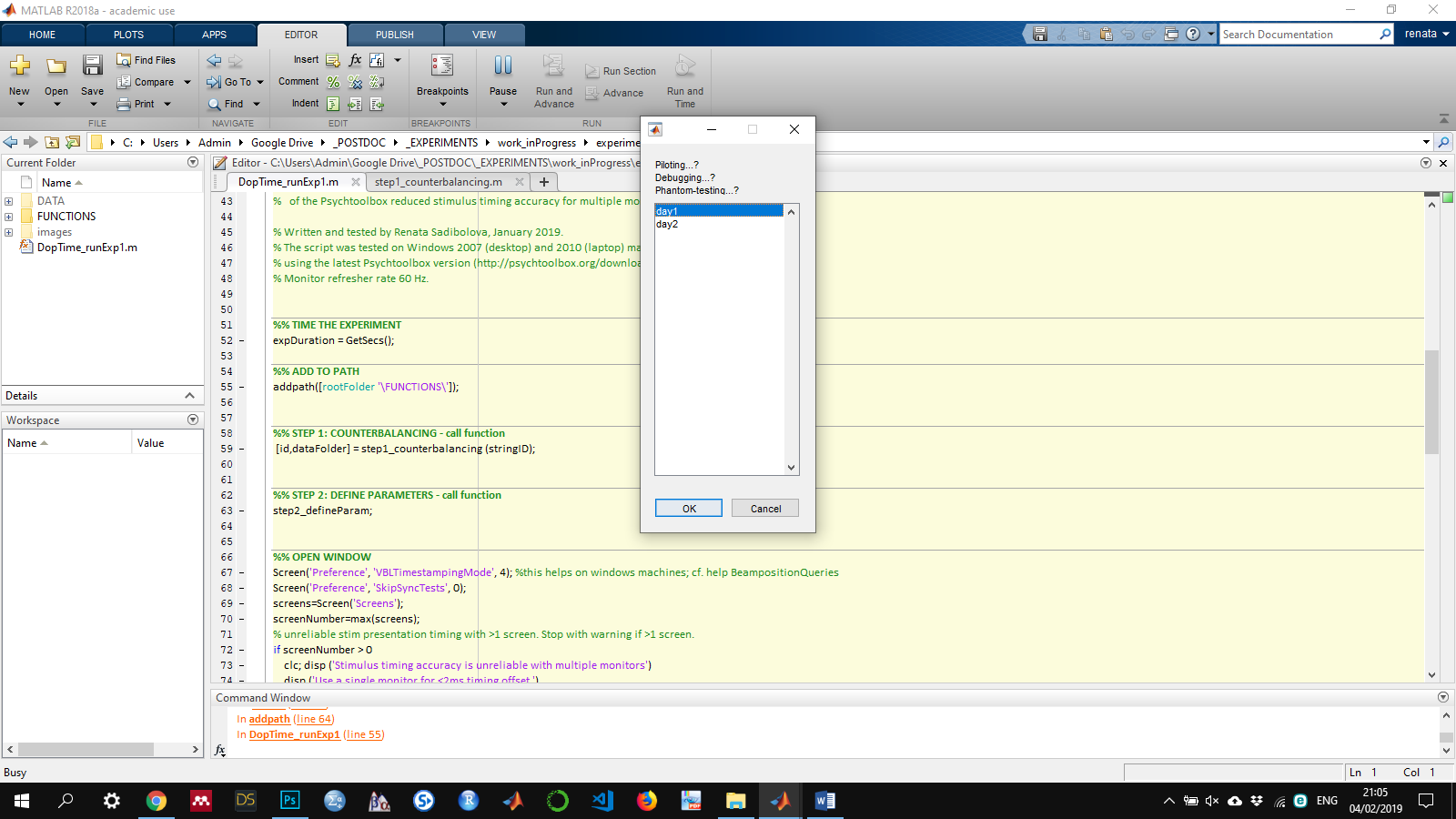
Type in the Matlab command window the function name followed by an actual patient ID and the ‘day1’ selection: **DATime\_runExp1 (‘subjID’,’day1’)**

* You’ll be asked to confirm the new patient before the new patient folder is created (their order number, used for counterbalancing, will be determined from a count of existing patient folders).
* If you type in *n*, the script will switch to test mode (OPTION 3 below).
* The data will be saved in the patient’s subfolder in the DAY 1.

**OPTION 2 (testing the patient on DAY 2)**

Type in the Matlab command window the function name followed by an actual patient ID and the ‘day2’ selection: **DATime\_runExp1 (‘subjID’,’day2’)**

* If the patient with this ID was tested on DAY 1, the script will automatically continue with no GUI. Otherwise, if there is no day1 data for the patient, a warning will pop up and the script will continue in the test mode (OPTION 3 below).
* The data will be saved in the patient’s subfolder in the DAY 2 folder

**OPTION 3 (any piloting and programme-testing)**

Type the function name in the Matlab command window followed by a string ID of your choice and the ‘testScript’ selection; e.g.:

**DATime\_runExp1 (‘patient\_pilot1’,’testScript’)**

or

**DATime\_runExp1 (‘XX\_healthyControl’,’testScript’)**

* You’ll be then asked to select which day to pilot:
* The final output file will be stored in newly created folder (its name = string ID of your choice). It will be saved in the TestScript folder

### Controller buttons

* Participants make their responses with controller buttons as shown below. The images are included in the instructions. The Response message screen (i.e. the left L and right S, or the right S and left L) are counterbalanced across participants. The actual left and right buttons on the controller were programmed for testing at Goldsmiths always as the keyboard keys **‘0’** and **‘.’**, respectively. The red button on the face of the controller is programmed as **‘SPACEBAR’** and it was used to begin new blocks.

The above can be changed to your pre-programmed controller keys in ***step2\_defineParam,*** **lines 9-11**.

Please be mindful:

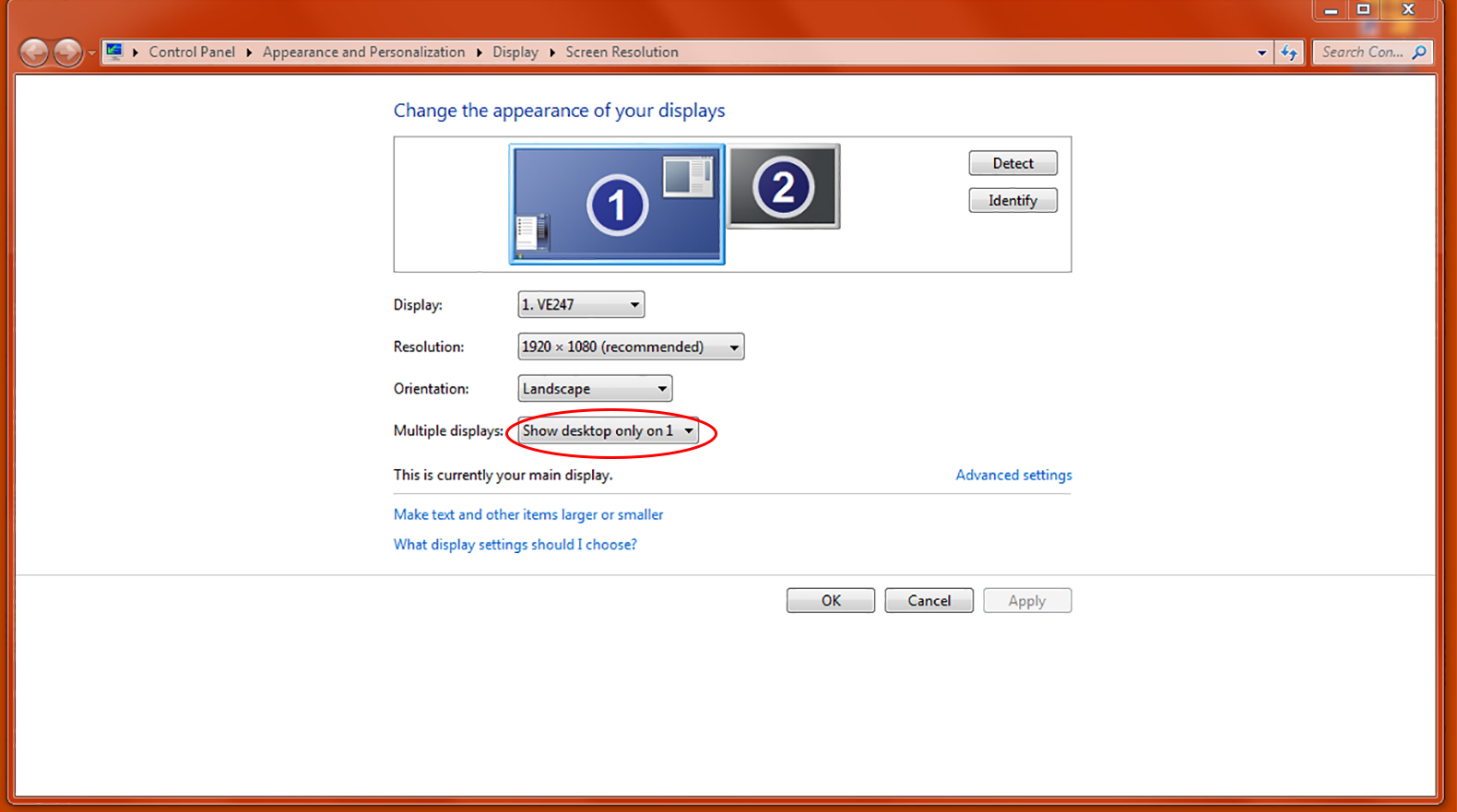
* Please make sure that the task script can record three inputs (i.e. initiate the block and record the short and long responses) and avoid that unused buttons, when accidentally pressed, do this.
* Please avoid the keys used by the function ***step3\_key5strokes*** (e.g. ‘5’ or **Esc**).
* To be extra sure, please check if the final data file contains the 0s (short responses) and 1s (long responses). This has been tested on our computers, and it should not be a problem. Find the responses in column 14 in finalData.outputTrain or finalData.outputTest.

### Key 5 pulses

* The recording of key 5 strokes was tested manually. **Please test the following:**
  + - The key 5 stroke is being recorded with a 100 Hz sampling rate.
    - We increased the buffer size to 500,000 events (this includes the press and release (i.e. 2 events per stroke) from a standard 10,000 (**lines 32** in ***step3\_key5strokes***), but we did not run a test recording with such a large amount of data. If problematic, we should aim to clear the buffer at the end of each block.
    - Recorded key 5 stroke data (saved as variable finalData.recordedKeys5) are the Matlab (Psychtoolbox) time stamps. These time stamps will have to be paired with the voltammogram time stamps.
  + They are recorded using KbQueue command in Psychtoolbox.
  + Note that we additionally record time stamps at each flip of the Psychtoolbox screen (several for each trial).
  + Both these commands implicitly use ‘GetSecs’ command. GetSecs uses the highest precision realtime clock available on operating system.
  + Please type in help GetSecsTest in the command window for more detailed information and to test the timing reliability on your computer. Please note that Psychtoolbox uses a number of ‘work-arounds’ to improve the timing if needed (e.g. see help BeampositionQueries).

### Timing accuracy

* To maintain good stimulus timing accuracy, the programme must **run on a single screen**. If it’s run on a laptop with an additional screen, please choose the ‘**show desktop only on 1’** option in the Windows Screen



* The stimulus timing accuracy should be double-checked on the new computer. Tested on our computers, we observed the offset of < 2ms. Please send us the pilot data files for checks. Some troubleshooting tips:
* To verify the timing accuracy in the output data files from the *Training* and *Testing phase* finalData.outputTrain and finalData.outputTest:
* Plot the difference between the programmed (column 1) and actual (column 4) stimulus intervals.
* Plot the difference between the programmed ISI 1 (column 2) and actual ISI 1 (column 5).
* Plot the difference between the programmed ISI 2 (column 3) and actual ISI 2 (column 6).
* The stimulus and ISI interval durations were selected to cover the integer number of flips for a 60 Hz monitor refresh rate (0.01667 s per frame). **Tip:** Check the refresh rate.
* Type in the Matlab command window: help BeampositionQueries.
* The programme currently uses the Screen('Preference', 'VBLTimestampingMode', 4).
* Change if needed in ***DopTime\_runExp1.m***, **line 71.**

### Stimulus size

* The stimulus (circle) size must be 2 – 4 degrees of visual field. **Please verify** this is the case on your monitor. The circle size can be changed in ***DopTime\_runExp1.m***, **lines 95-96.**

**Tip:** Use trigonometry to compute the circle size (angle = 2 to 4 °VF) in pixels on your monitor using its distance from a participant (adjacent line). If you send us the diameter length of the circle as it appears on the screen and the distance between the screen and the participant, we can verify the visual angle

## 3.1. The task script: details