

Abertay University

# CulTra Documentation

Pilot-Testing CulTra: Crowd-Sourcing Research Games for the Study of Cultural Transmission



Abertay  
University



THE CARNEGIE TRUST  
FOR THE UNIVERSITIES OF SCOTLAND

Nikolay Panayotov  
Vera Kempe

April 2019, updated July 2019

# 1 TABLE OF CONTENTS

---

|       |   |    |
|-------|---|----|
| 2     | About the Project Code and This Documentation ..... | 3  |
| 2.1   | Included Project Files .....                        | 3  |
| 2.1.1 | db .....  | 3  |
| 2.1.2 | private .....                                       | 3  |
| 2.1.3 | public_html/binary-language.....                    | 3  |
| 2.1.4 | public_html/buzzer-language .....                   | 3  |
| 2.1.5 | public_html/two-colour-language .....               | 3  |
| 3     | Experiment Design .....                             | 4  |
| 3.1   | Materials .....                                     | 5  |
| 3.1.1 | Items .....   | 5  |
| 3.1.2 | The Implementation .....                            | 5  |
| 3.2   | Buzzer Language Study .....                         | 6  |
| 3.3   | Two Colour Language.....                            | 9  |
| 4     | Database Design.....                                | 12 |
| 4.1   | Database Relationships.....                         | 12 |
| 4.2   | Table Outlines .....                                | 12 |
| 4.2.1 | Tables for Buzzer Language.....                     | 12 |
| 4.2.2 | Tables for Two-Colour Language .....                | 12 |
| 4.3   | Tables Structure .....                              | 13 |
| 4.3.1 | transmission_sessions.....                          | 13 |
| 4.3.2 | nodes.....  | 14 |
| 4.3.3 | transmissions .....                                 | 15 |
| 5     | Code Organisation.....                              | 17 |
| 5.1   | Directory and Files Structure .....                 | 17 |
| 5.1.1 | Entry part: outside the 'study' folder.....         | 17 |
| 5.1.2 | Main part: inside the 'study' folders.....          | 17 |
| 5.1.3 | Resources .....                                     | 17 |
| 5.1.4 | Pages / HTML .....                                  | 17 |
| 5.1.5 | CSS.....  | 17 |
| 5.1.6 | JavaScript .....                                    | 18 |
| 5.1.7 | PHP .....   | 18 |
| 5.1.8 | Note on the 'tree' folders .....                    | 18 |
| 6     | Implementation Design and Architecture .....        | 19 |
| 6.1   | Setting Up.....                                     | 19 |

|       |   |    |
|-------|---|----|
| 6.2   | Initialising a Session .....  | 19 |
| 6.3   | Setting Experimental Conditions and User Data.....                      | 19 |
| 6.4   | Handling the Allocation of Nodes to Sessions Concurrently .....         | 20 |
| 6.5   | The Study State Machine .....   | 20 |
| 6.6   | Study Sections.....   | 21 |
| 6.6.1 | Information Screens and Questionnaires .....                            | 21 |
| 6.6.2 | Comprehension Screens .....   | 21 |
| 6.6.3 | Production Screens .....  | 21 |
| 7     | Resources.....  | 22 |
|       | Used code: .....  | 22 |
|       | Appendix: Exact Information and Instructions Given to Participants..... | 23 |
|       | Buzzer Language Study .....   | 23 |
|       | Two Colour Language Study.....  | 27 |

## 2 ABOUT THE PROJECT CODE AND THIS DOCUMENTATION

---

The Project, codenamed *CulTra*, is a series of web-based psychology experiments attempting to study cultural evolution and cultural transmission through an automatic online framework. The project was led by Prof Vera Kempe with Nikolay Panayotov building the necessary technical solutions. The present documentation outlines the design and implementation of the experiments from a technical standpoint. Familiarity with the project at large is assumed.

All project files referred to here are built with HTML5 and related web technologies, such as JavaScript for client-side code and PHP for server-side code.

### 2.1 INCLUDED PROJECT FILES

#### 2.1.1 db

Contains an .sql file describing the database structure of the two experiments.

#### 2.1.2 private

Contains an example config.ini file used to access your database. This file tends to contain login details so it should be kept away from public access folders.

#### 2.1.3 public\_html/binary-language

The file in this folder simply redirects the user to one of the two experiments by a simple random selection.

#### 2.1.4 public\_html/buzzer-language

This folder contains the source files for the 'buzzer' experiment that employs sounds for the artificial language. Details in the next section.

#### 2.1.5 public\_html/two-colour-language

This folder contains the source files for the 'colour' experiment that employs coloured dots for the artificial language. Details in the next section.

### 3 EXPERIMENT DESIGN

There are two main experiments that were conducted: one experiment involving a 'buzzer language' with sound and manipulating the passing requirement for the adaptive training (easier or harder); and another experiment involving a 'two colour language' with coloured dots and manipulating the method of presentation (stepwise-fading or holistic-static). Both experiments have two different types of training blocks (comprehension or production), each one with 8 trials (one for each word/meaning in the language). Completing a trial correctly gives the participant a point for that block; getting enough points in the block (based on a minimum passing requirement) results in a pass, which awards the participant with a 'star point'. If the participant does not get enough points in the 8-trial block, the score is reset and they move on to the other type of block without gaining a 'star point'. The two types of training blocks alternate until the participant has collected 2 'star points'. Successfully collecting 2 'star points' leads the participant to Production Testing where their output becomes the input for the next participant in the chain. This is followed by a Self-Testing Comprehension block that simply tests the participant on their own output by selecting 4 of their inputs from the previous screen. This completes the experiment and participants are asked for their feedback before receiving their debrief.

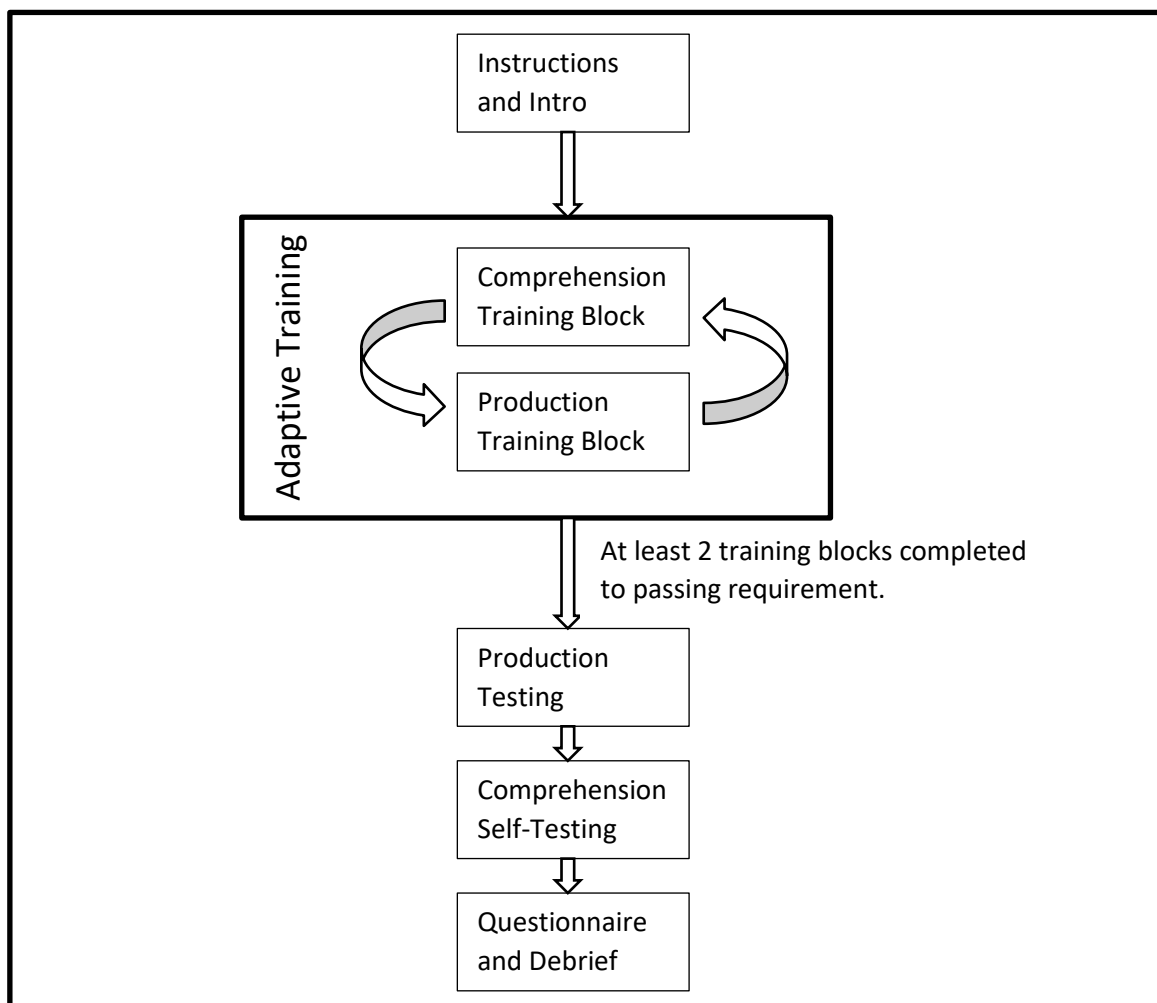










Figure 1. A diagram of the general experiment procedure.

## 3.1 MATERIALS

### 3.1.1 Items

The languages for both experiments consisted of 8 meanings as outlined in Table 1.

*Table 1. All items used and their image and properties.*

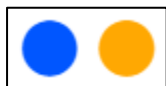
|       |   |   |   |   |   |   |   |   |
|-------|---|---|---|---|---|---|---|---|
| Image |  |  |  |  |  |  |  |  |
| Name  | a   | b   | c   | d   | e   | f   | g   | h   |
| Size  | Big   | Big   | Small   | Small   | Big   | Big   | Small   | Small   |
| Shade | Light   | Light   | Light   | Light   | Dark  | Dark  | Dark  | Dark  |
| Shape | Rounded   | Spiky   | Rounded   | Spiky   | Rounded   | Spiky   | Rounded   | Spiky   |

Each meaning is associated with a word, which is a binary sequence, where one experiment represents that as audible high-pitch and low-pitch buzzers, and the other experiments represents the binary sequences with coloured dots, either blue or orange (Figure 2).

In every block of the experiments the words are randomised across 8 trials so their order of presentation is different each time.

#### Audio for Buzzers:

Two 500ms long, sine waves at 293.66Hz (note D4) and 440Hz (note A4).



*Figure 2. Appearance of the Coloured Dots.*

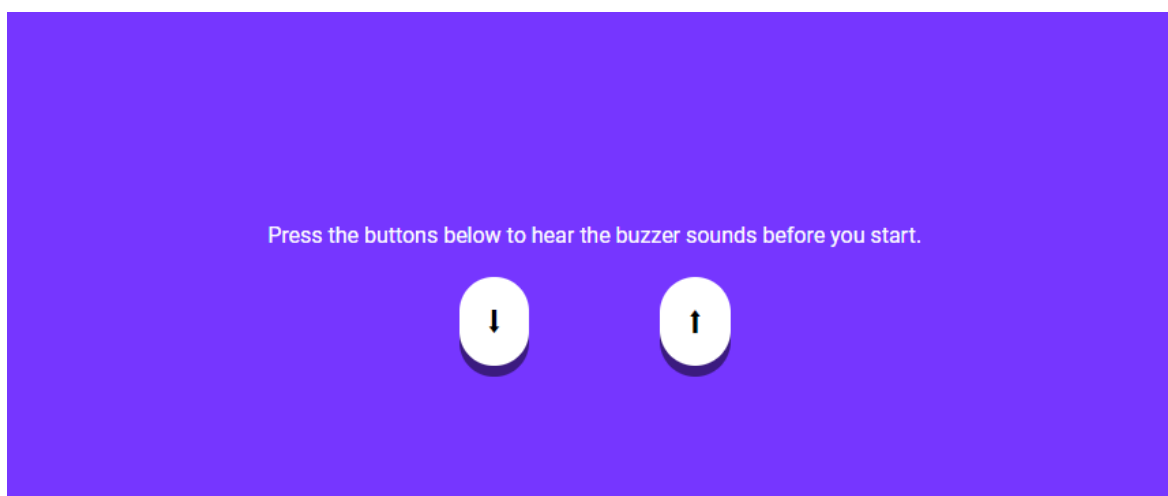
### 3.1.2 The Implementation

The two experiments, titled 'Buzzer Language' and 'Two Colour Language' were implemented for the web browser using HTML5 technologies. Participants could be recruited online via a web link and they had the opportunity to run the study on a variety of devices, including desktops, laptops, phones and tablets.

### 3.2 BUZZER LANGUAGE STUDY

The main manipulation of this study is the training condition, particularly how difficult the adaptive training block is. This is manipulated by modifying the required passing score to earn a ‘star point’ from a particular training block. This can be either 5 points of 8 (harder condition) or 3 points of 8 (easier condition). Passing 2 training blocks earns the participant 2 star points that allows them to continue to the testing blocks.

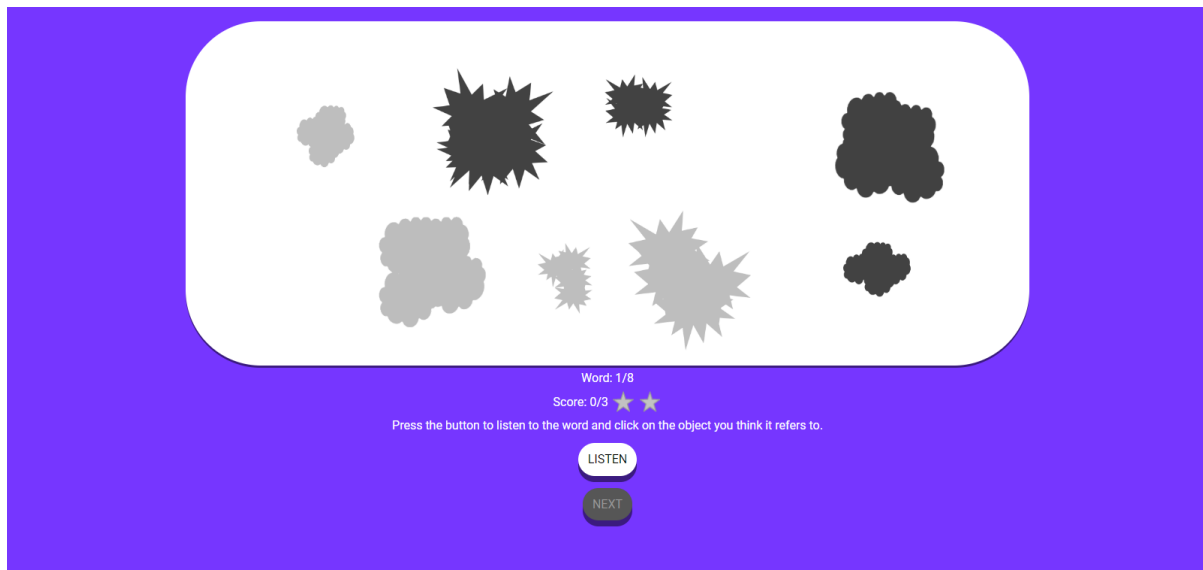
Before each block in the Buzzer Language Study, participants are shown two buttons labelled with an upward or downward arrow (Figure 3) and are asked to press them to hear their corresponding sounds (high pitch and low pitch respectively). This allows participants to familiarise themselves with the interface they would be using throughout the experiment and lets them hear the sounds and adjust their volume accordingly before starting. The left/right order of the buttons is randomly counterbalanced across participants.



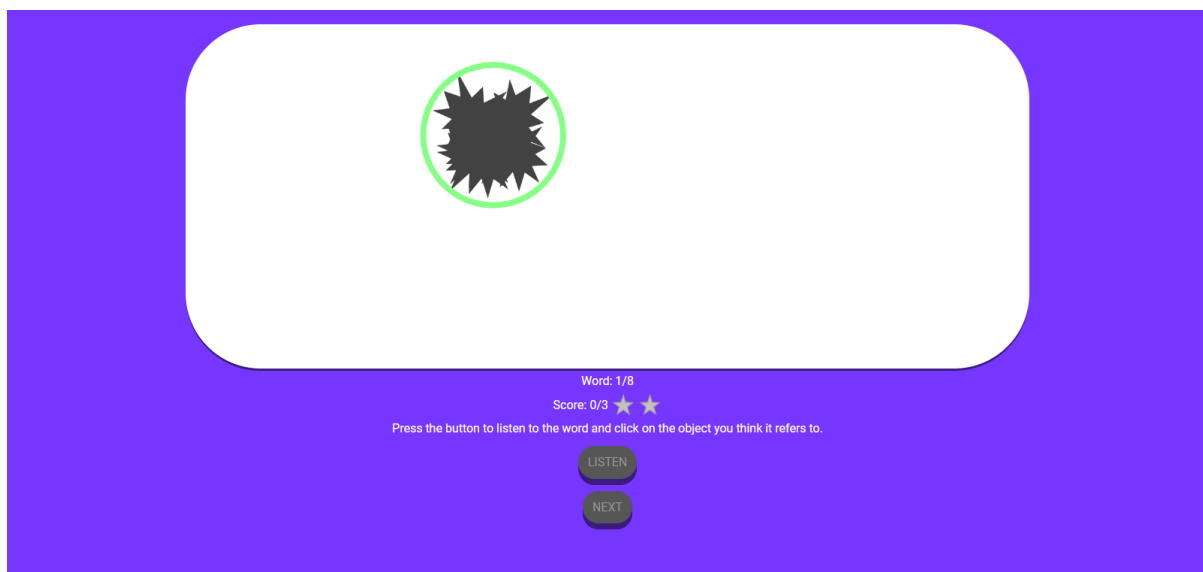
*Figure 3. Screenshot of buzzer readiness trial.*

The first block participants encounter is a Comprehension Training block, where a white canvas on the screen displays all the 8 meanings (objects) in the language in a dispersed manner (Figure 4). The participants are prompted to press the “Listen” button to hear this trial’s word and then select one of the 8 object they think it refers to. After pressing the button, participants hear a binary sequence of high and/or low buzzer sounds (as described in the Materials section) – each buzz is 500ms long and there is a 500ms silence pause between each buzz in a word. After the sound sequence finishes playing, participants are allowed to click or press on the canvas to point to the word meaning they think the sound referred to. If they guess correctly, the object is surrounded by a green circle and participants earn a point and their score on the screen updates. If they guess wrongly, the object they selected is surrounded by a red circle and the correct object is surrounded by a green circle – participants’ score remains unchanged. Immediately after that all objects, except the correct, circled in green, object fade away. Once the only object visible on screen is the correct answer (Figure 5), the sound sequence for the meaning plays again as before to remind participants of the word. This completes the trial, the objects reappear on the screen in a different spatial configuration and participants move on to listening to the next word and trying to guess its meaning. After 8 trials going through all the words of the languages, participants move on to the next block.

If during the training block participants score (guess correctly) enough words (based on their training condition), they earn a star point. Otherwise, they continue to the next block without gaining a star point.



*Figure 4. Comprehension Block for the Buzzer Language Study.*



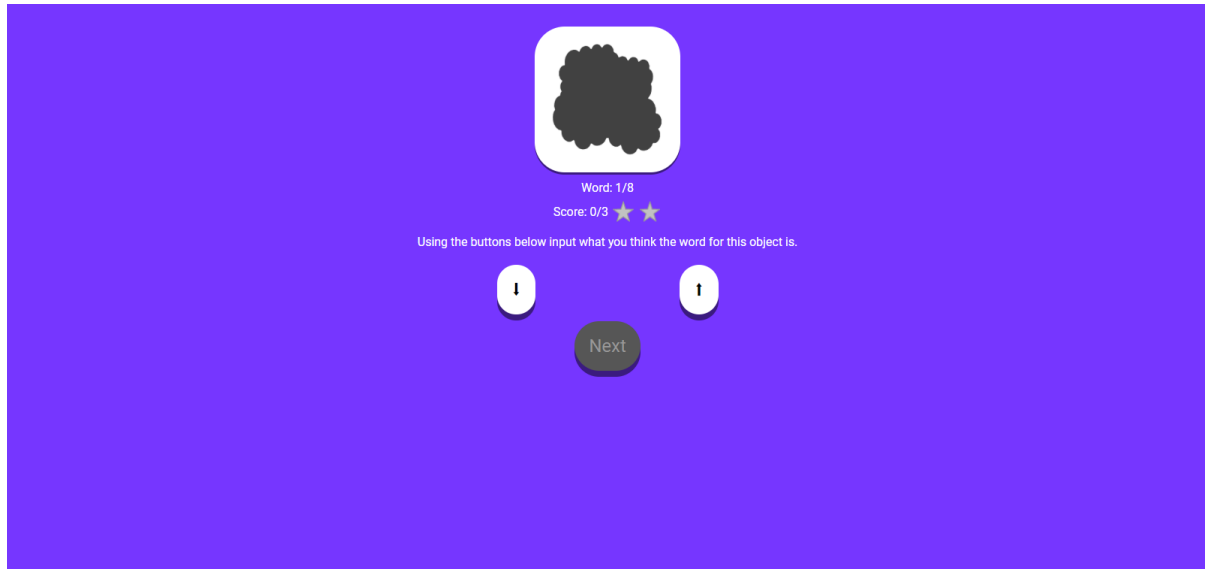
*Figure 5. Showing correct response feedback after selection in the Buzzer Language Study.*

The Comprehension Training block is followed by the Production Training block. In this block the meanings/object images appear one by one and each time participants are asked to use the on-screen buttons to instead input the sequence of sound themselves (Figure 6). When pressing a button, participants hear the 500ms buzz piece and can string multiple ones together. Once they think they are done, they can press the “Next” button. Following this they are told whether their input matched the actual word or not with the prompt “Correct!” or “Wrong!”, after which they



hear the correct sound sequence as before. The participants repeat this for all 8 words in the language, gaining a point every time they are correct.

If during the training block participants score enough points (based on their training condition), they earn a star point. Otherwise, their adaptive training procedure continues back to the Comprehension Training block and keeps alternating between comprehension and production training until the participant has collected 2 star points.



*Figure 6. Production Block for the Buzzer Language Study.*

Successfully collecting 2 start points completes the adaptive training procedure and the Production Testing block is initiated (after a short chime to refocus participants). This block follows exactly the same procedure as the Production Training, except no feedback is given and the 'correct' sequences are never played. Participant's input in this block is used as the language to learn for the next participant in the chain (unbeknownst to the participant). This is why participants are not allowed to input the same sequence twice for different objects and are told to retry until they produce a unique sequence for each word.

The Production Testing block is followed by the Comprehension Testing block. This is identical to the Comprehension Training block, except participants are told that they will hear their own sound sequences that they created in the previous block. No feedback is given about the correctness of the choices and it goes for only 4 trials.

Finally, participants are asked how they found out about the study and to rate their enjoyment of the experiment on a 5-point scale. They are also provided with a comment box to include any feedback they would like before being debriefed. The debrief screen also shows participants the percentage of correct trials they did on their Comprehension Testing block.

The exact information and instructions provided to the participants throughout the study are shown in the appendix.

### 3.3 TWO COLOUR LANGUAGE

This study uses coloured dots (a sequence of blue and orange dots) to represent the binary language and the main manipulation is the method of presentation of the coloured dots, which can be either stepwise-fading or holistic-static. The stepwise-fading condition reveals the dots one by one by fading away the previous dot after placing the next one – the interval of time between each dot onset is 1000ms, while the fading animation of the old dot is 400ms and is initiated simultaneously with the onset of the new dot. The holistic-static condition simply reveals all the dots of the word at the same time and keeps them on screen for  $X \times 500\text{ms}$ , where  $X$  stands for the length of the binary string representing the word.

The first block participants encounter is a Comprehension Training block, where a white canvas on the screen displays all the 8 meanings (objects) in the language in a dispersed manner (Figure 7). The participants are prompted to press the “Show Word” button to see this trial’s word and then select one of the 8 object they think it refers to. After pressing the button, participants see the binary sequence in the form of a sequence of blue and orange dots on an input field below the canvas. The exact method of presentation is different for the two main conditions as described above. After seeing the sequence, the participants are allowed to click or press on the canvas to point to the word meaning they think the sequence referred to. If they guess correctly, the object is surrounded by a green circle and participants earn a point and their score on the screen updates. If they guess wrongly, the object they selected is surrounded by a red circle and the correct object is surrounded by a green circle – participants’ score remains unchanged. Immediately after that all objects, except the correct, circled in green, object fade away. Once the only object visible on screen is the correct answer (Figure 8), the sequence for the meaning is shown again as before to remind participants of the word. This completes the trial, the objects reappear on the screen in a different spatial configuration and participants move on to seeing the next word and trying to guess its meaning. After 8 trials going through all the words of the languages, participants move on to the next block.

If during the training block participants score (guess correctly) 3 words, they earn a star point. Otherwise, they continue to the next block without gaining a star point.

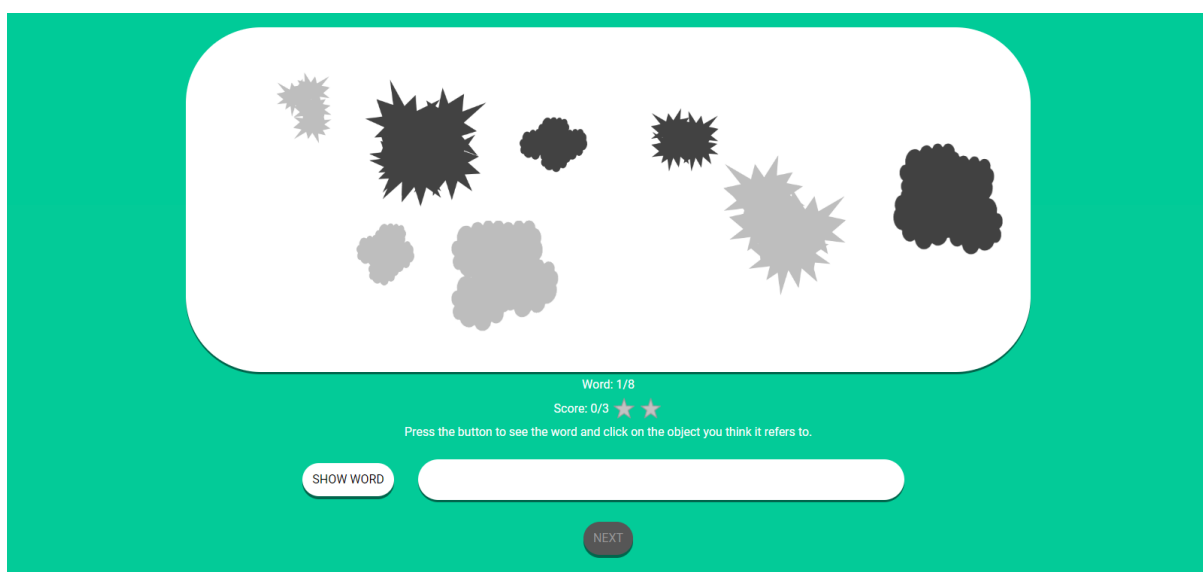
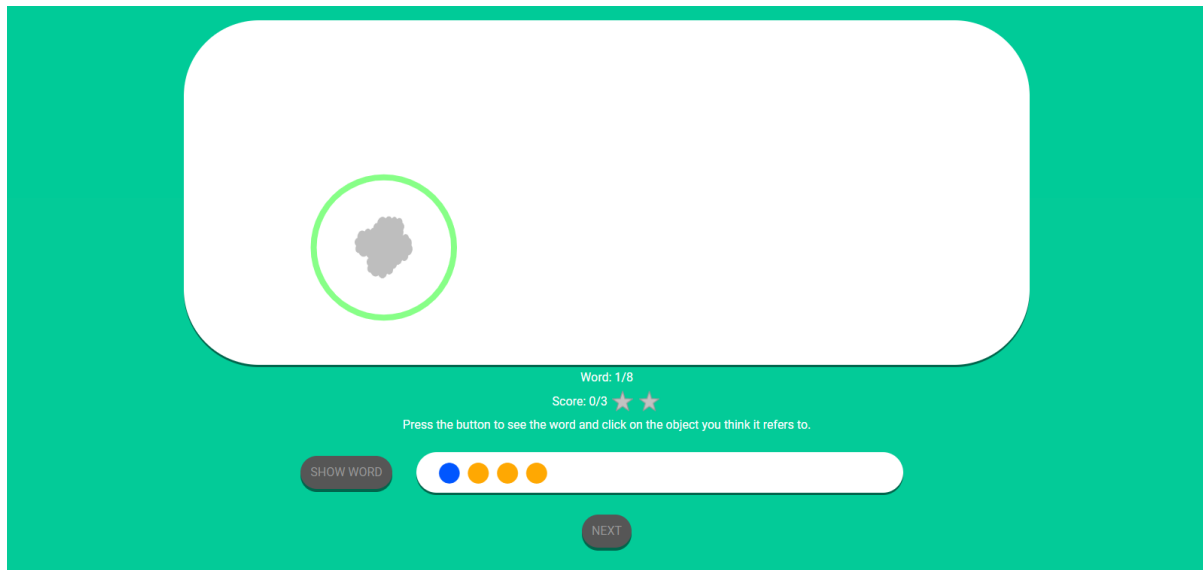


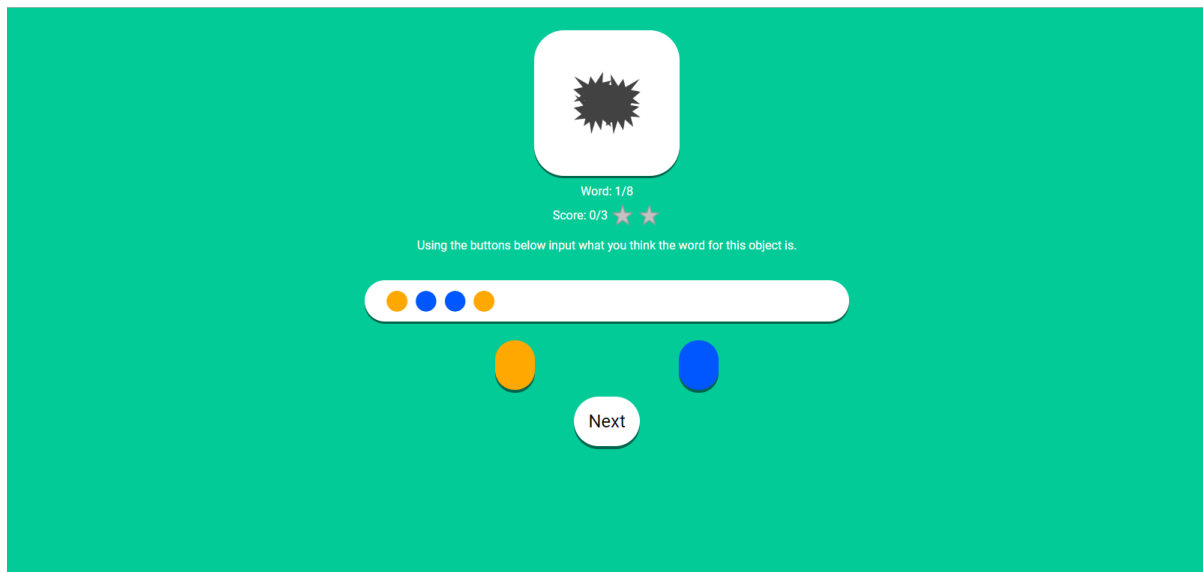
Figure 7. Comprehension Block for the Two Colour Language Study.



*Figure 8. Showing correct response feedback after selection in the Two Colour Language Study.*

The Comprehension Training block is followed by the Production Training block. In this block the meanings/object images appear one by one and each time participants are asked to use the on-screen buttons to instead input the sequence of coloured dots themselves (Figure 8). The left/right order of the coloured buttons is randomly counterbalanced across participants. When pressing a button, participants see a new dot appear in the input field on the screen and can string multiple ones together. In the stepwise-fading condition any previous dot disappears after a new one is entered, but in the holistic-static condition all dots remain until submission. Once participants think they are done, they can press the “Next” button. Following this they are told whether their input matched the actual word or not with the prompt “Correct!” or “Wrong!”, after which they see the correct coloured dot sequence as before. The participants repeat this for all 8 words in the language, gaining a point every time they are correct.

If during the training block participants score 3 points, they earn a star point. Otherwise, their adaptive training procedure continues back to the Comprehension Training block and keeps alternating between comprehension and production training until the participant has collected 2 star points.



*Figure 9. Production Block for the Two Colour Language Study.*

Successfully collecting 2 start points completes the adaptive training procedure and the Production Testing block is initiated. This block follows exactly the same procedure as the Production Training, except no feedback is given and the 'correct' sequences are never shown. Participant's input in this block is used as the language to learn for the next participant in the chain (unknown to the participant). This is why participants are not allowed to input the same sequence twice for different objects and are told to retry until they produce a unique sequence for each word.

The Production Testing block is followed by the Comprehension Testing block. This is identical to the Comprehension Training block, except participants are told that they will see their own dot sequences that they created in the previous block. No feedback is given about the correctness of the choices and it goes for only 4 trials.

Finally, participants are asked how they found out about the study and to rate their enjoyment of the experiment on a 5-point scale. They are also provided with a comment box to include any feedback they would like before being debriefed. The debrief screen also shows participants the percentage of correct trials they did on their Comprehension Testing block.

The exact information and instructions provided to the participants throughout the study are shown in the appendix.

## 4 DATABASE DESIGN

---

### 4.1 DATABASE RELATIONSHIPS

Initially, the database starts with a couple entries in the *nodes* table, which represents a tree data structure. Each node (or entry) in the table records the ID of its parent node, which allows us to trace the continuity of the chain. Each new experiment session is assigned a node which is at the end of the chain (or in other words, a leaf in the tree). This sets the status of the node from 'free' to 'taken' (occupied) and prevents other sessions from taking it. A completed experiment session creates a new child node in the tree, parented by the taken session. The generated transmissions data from the session in the *transmissions* table are then associated with the node as well. The *transmission\_sessions* table serves to link the nodes to their associated transmissions data and keeps track of the experiment's progress.

### 4.2 TABLE OUTLINES

The buzzer and colour tables are essentially identically structured and simply refer to different experiments.

#### 4.2.1 Tables for Buzzer Language

**nodes** – list of nodes and their properties

**transmissions** – list of transmissions and their details

**transmission\_sessions** – list of sessions (participants) and their details

#### 4.2.2 Tables for Two-Colour Language

**colour\_nodes** – list of nodes and their properties

**colour\_transmissions** – list of transmissions and their details

**colour\_transmission\_sessions** – list of sessions (participants) and their details

## 4.3 TABLES STRUCTURE

### 4.3.1 transmission\_sessions

**`session\_number` int(11) UNSIGNED**

A unique identifier for each experiment session. It serves as a link between the other tables.

**`prolific\_id` text**

**`prolific\_session` text**

Two variables used by the online participant recruitment platform Prolific.

**`completion\_code` char(10)**

The completion code presented to the participant at the end of the study.

**`node\_id` int(11) UNSIGNED**

The new node ID associated with this session, it remains 0 (zero) until the session is complete and the new node is actually created.

**`tree` int(11)**

This is the ID of the particular tree or chain for the current node. Conditions are determined by the tree number of the session and its corresponding transmissions.

For the buzzer language study:

If tree <= 6

Pass requirement = 5 correct trials per block;

Else

Pass requirement = 3 correct trials per block;

For the colour language study:

Pass requirement = 3 correct trials per block always.

If tree <= 6

Permanent condition;

Else

Fading condition;

**`parent\_id` int(10) UNSIGNED**

The node ID of the parent node for this session. The referenced node is where the previous language binary sequences come from.

**`generation` int(11) UNSIGNED**

The corresponding position in the chain (or the evolution generation).

**`buttons` varchar(9)**

This is the counterbalancing of the location of the buttons in the experiment. The values can be “leftHigh” or “rightHigh”. It describes on which side the ‘high tone’ or ‘blue dot’ appear respectively (for two-colour language simply replace “High” with “Blue”).

**`progress` varchar(32)**

The screen name the session has reached. If the session is completed, this should say END.

**`start\_timestamp` timestamp**

The time when the session started.

**`end\_timestamp` timestamp**

The time when the session was completed.

**`enjoy` tinyint(3) UNSIGNED**

Result of the questionnaire asking if the participant had fun with the study. A Likert-type scale between 1 and 5.

**`outreach` varchar(30)**

Stores the participant’s input about how they found out about the experiment.

**`comment` varchar(500)**

Stores the feedback participants can give optionally.

**`browser` tinytext**

This string usually shows the type of browser the participant was using.

#### 4.3.2 nodes

**`timestamp` timestamp**

The time when the node was created.

**`node\_id` int(11) UNSIGNED**

A unique identifier of the tree node.

**`tree` int(11)**

See ‘tree’ under the *transmission\_sessions* table.

**`parent\_id` int(10) UNSIGNED**

The ID of the parent node to this node.

**`generation` int(11) UNSIGNED**

The corresponding position in the chain (or the evolution generation).

**`session\_number` int(10) UNSIGNED**

The session associated with creating this node.

**`node\_type` varchar(11)**

This can be various types, but the most important are 'terminal' (leaf) and 'nonterminal' (stem). Only terminal nodes can be taken by new sessions.

**`status` varchar(16)**

Describes the availability of the node, either 'free' or 'taken'. A new session can only take a free node to avoid synchrony issues.

**`expires` timestamp**

A timestamp indicating when a 'taken' node can turn back into a 'free' node IF the session that has taken it has been inactive until that time. This is to prevent unfinished sessions from blocking new sessions.

### 4.3.3 transmissions

**`task\_id` int(11) UNSIGNED**

A unique identifier for the trial.

**`timestamp` timestamp**

The time of completing the trial.

**`session\_number` int(11) UNSIGNED**

The session that produced the data.

**`node\_id` int(11) UNSIGNED**

The node associated with the data (remains 0 until the experiment is complete)

**`tree` int(11)**

The tree (or chain) associated with the data.

**`parent\_id` int(10) UNSIGNED**

The ID of the parent node.

**`generation` int(11) UNSIGNED**

The corresponding position in the chain (or the evolution generation).

**`section` varchar(32)**

Identifying if this is a comprehension or production, training or testing trial, based on the current screen.

**`cycle` int(11)**

The number of the training cycle in the experiment.



**`item\_order` tinyint(4) UNSIGNED**

The order of the trial in the experiment.

**`object` char(1)**

The object the trial was referring to (as defined in the procedure section).

**`target` varchar(20)**

The 'expected' binary sequence for the trial, such 'aabab'.

Sequences are coded to follow a pattern of 'a' and 'b', where:

Buzzer Language

a - high tone

b - low tone

Two-Colour Language

a - blue

b – orange

Alternatively, in the comprehension screen, this is the name of the object that the participant should have selected (identical to the *object* column).

**`input` varchar(20)**

The binary sequence that the participant actually inputted for this trial. Or in the comprehension screen, the name of the object that was selected.

**`correct` tinyint(1)**

A Boolean describing whether the input matched the target.

**`edit\_distance` float**

A calculated Levenshtein edit distance between the target and the input string.

## 5 CODE ORGANISATION

---

### 5.1 DIRECTORY AND FILES STRUCTURE

#### 5.1.1 Entry part: outside the 'study' folder

In the *public\_html/buzzer-language* and *public\_html/two-colour-language* are the entry points of the two experiments respectively. These folder are the entry points for participants. Here there are some introductory pages and an ethics form, which is completely client-side and does not record any data until participants accept the terms and move on to the main part. URL variables are passed on to the URL of the main part.

#### 5.1.2 Main part: inside the 'study' folders

Participants move on to this part of the respective experiment after agreeing to the ethics form. This results in the creation of a new session from *create\_new\_session\_for\_index.php*, which connects to the database through *database\_connection\_for\_index.php* and creates a new entry in the sessions table. The URL variables are extracted with *get\_url\_data.php* and conditions are set and recorded. The main study control flow is organised in a state machine in *index.php*.

#### 5.1.3 Resources

There are two resource folders, one for images and one for audio.

Images contains the pictures used for the objects of the language's meaning space and some interface images.

Audio contains the sounds used for the buzzers.

#### 5.1.4 Pages / HTML

These are all the pages used in the experiment in HTML format. They are presented to the user from the experiment main control flow in *index.php*.

#### 5.1.5 CSS

There are two styling sheets for the entire experiment:

*normalize*: standardises the default styling across browsers and is used on every page.

*global*: is specific to the experiment, but also applies to all pages.

### 5.1.6 JavaScript

The JavaScript files contain specific code to be run by particular sections of the experiment. Details are discussed in the next section. *jQuery* is also used.

### 5.1.7 PHP

This is the server-side code of the program, which is entirely written in PHP. Details of the implementation and specific files are discussed in the next section.

### 5.1.8 Note on the ‘tree’ folders

Besides the ‘*study*’ folder, the two experiments also contain a ‘*tree*’ folder. This contains a separate interface from the experiment which visualises the tree data structure generated as participants fill up the chains. The visualisation is built in the JavaScript library D3.js and communicates with the database in a similar way to the main study. The visualisation is an extra feature to help the researchers track the progress of their study overall and is never seen by participants. It is not the focus of this documentation.

## 6 IMPLEMENTATION DESIGN AND ARCHITECTURE

---

### 6.1 SETTING UP

1. Use *binary\_language\_database.sql*, in the *db* folder, to create the database on your server.
2. In the *study* folder, modify the function **connect()** in *database\_connection\_for\_index.php* and *database\_connection.php* to direct the **parse\_ini\_file** function to an appropriate *config.ini* file (ideally, outside your server's *public\_html* files folder) containing your database's credentials. Plug those in the **mysqli** constructor. An example config file is located in the *private* folder.
3. Upload the updated *CulTra* experiment folder to your server. Make sure your modified paths in the database connection files match your server's file directory.
4. Check that everything is working.
5. All done!

### 6.2 INITIALISING A SESSION

The study officially commences when the user requests the *index.php* file in the study folder and the program begins creating records and storing data. The first thing done by *index.php* is to check if a session has been created by checking an 'exists' property; if not, any potential URL variables are grabbed by *get\_url\_data.php* and a new session is initiated through the *create\_new\_session\_for\_index.php*.

The *create\_new\_session\_for\_index.php* does several important tasks:

- Selects experimental conditions, based either on what *get\_url\_data.php* gathers;
- Creates a new session record in the sessions table and the associated Prolific table;
- Initialises and resets all experiment-wide variables to be discussed in other sections of this documentation.
- Sets the initial state of the program as 'START', which will be used by the study state machine. This is stored in ***\$\_SESSION['state']***.
- Assigns a free node to the session.

### 6.3 SETTING EXPERIMENTAL CONDITIONS AND USER DATA

Experimental conditions, as described previously, are set in the *create\_new\_session\_for\_index.php* either automatically based on the session number following a simple alternation pattern or explicitly by a custom URL of the study. The file *get\_url\_data.php* extracts specific conditional settings, as well as external data (such as Prolific variables), and stores them in the session, which is subsequently recorded in the session record in the sessions table. Other experimental conditions can easily be added by including them in the same way as the others.

## 6.4 HANDLING THE ALLOCATION OF NODES TO SESSIONS CONCURRENTLY

The file *create\_new\_session\_for\_index.php* queries the database for nodes that have their *status* property set to 'free'. If none are free, it returns a message saying the server is busy. If a free node is found it is assigned to the session and set to 'taken', allowing the experiment to begin. To avoid the session 'hogging' the node, even if the session never completes the experiment, an 'expiration' timestamp is set to the node 5 minutes in the future. Every time a participant queries the database (i.e. they interact with the experiment) a piece of code in *tree\_management.php* is ran. This code, expressed in the function **refreshSession()**, checks if any nodes have expired and sets them back to free if they have. In addition to this, it refreshes the current session's possession of its respective node by resetting the 'expiration' timestamp to 5 minutes in the future from that point. In other words, if the session does not show any action for 5 minutes, it will lose possession of the node, which might be taken by another session.

If the session reaches the end of the experiment, its assigned node becomes nonterminal and creates a new terminal node in the tree. All data generated from the session is labelled with the new node's ID and the next session can draw from it to achieve 'the transmission'.

## 6.5 THE STUDY STATE MACHINE

The main role of *index.php* is to manage the experiment's state machine – this is implemented with a simple switch statement.

Each state (or section) in the experiment has a certain number of trials (items) that the participants must pass before moving to the next section, called '**item\_total**'. When a trial is passed, a counter called '**item\_order**' is incremented by 1. When **item\_order** exceeds **item\_total**, *index.php* is refreshed by the client page and the state machine progresses to the next state.

The function **stateTransition(\$current\_state, \$next\_state)** in *index.php* determines the HTML page to be loaded for the current state and the name of the next state to go to once the total number of trials (items) in the current state have been passed. When a state is updated, this is also recorded in the session record in the sessions table, under the '**progress**' field. This field is used to keep track of the state machine.

The adaptive training is achieved through a 'training loop' in the state machine, where the session variables **training\_length\_condition** and **training\_score** are compared to determine whether the participant has passed enough training screens to progress to testing.

At the end of the state machine, the study completion code is also created and recorded, which is then displayed to the participant in the final HTML page.

## 6.6 STUDY SECTIONS

### 6.6.1 Information Screens and Questionnaires

Information screens and questionnaire forms are simply treated like regular sections that have only 1 total trials/items which is passed when the 'next' or 'submit' button is pressed: this is accomplished through the *next\_item.php* file. As usual, this triggers a page refresh and running the state machine again to transition to the appropriate next screen.

### 6.6.2 Comprehension Screens

Comprehension screens comprise of an HTML5 canvas that draws, in random 'shuffled' positions, all 8 objects of the experiment, making sure that they do not overlap with each other. The participant can click on the objects directly (after hearing/seeing the word) to complete a trial. There are a few circle collision functions to manage these behaviours.

There is an additional client-side state machine that handles the exact order of events, as described in the procedure section. The presentation of the language binary sequences is times with a series of **setTimeout()** functions.

Client-side behaviour is handled by *comprehension\_training.js*, while the server-side is handled by the *comprehension\_training.php*. The server receives the interaction information from the client, stores it, and serves the next trial's information.

The self-testing screen follows a near identical procedure, but does not give feedback.

### 6.6.3 Production Screens

The *generate\_buttons.php* file produces the interaction buttons in the right order from left to right that the participant uses to input their own binary sequences. Objects are presented in an *img* element one by one, as they are served from *production\_training.php*.

Client-side behaviour is handled by *production\_training.js*, while the server-side is handled by the *production\_training.php*. The server receives the interaction information from the client, stores it, and serves the next trial's information.

The testing screen follows a near identical procedure, but does not give feedback.

## 7 RESOURCES

---

### USED CODE:

Normalize.css: <https://github.com/necolas/normalize.css>

jQuery 3.1.1: <https://jquery.com/>

Using PHP with MySQL - the right way (Lionite): <https://www.binpress.com/tutorial/using-php-with-mysql-the-right-way/17>

D3.js: <https://d3js.org/>

# APPENDIX: EXACT INFORMATION AND INSTRUCTIONS GIVEN TO PARTICIPANTS

---

## BUZZER LANGUAGE STUDY

### The Binary Buzzer Language

#### Can you understand and use a language made up of only two sounds?

Thanks for considering participation in our language learning game! We are really glad you are here!

Our research group is trying to understand whether humans can learn a very simple language consisting of only two sounds - a high tone and a low tone.

Turns out this is not that easy so to better understand why we are trying to track the learning process of many participants.

To do so, it is really important for us that once you start you continue until you reach the learning goal which will be assigned to you.

It may take between 15 to 30 min or even a bit more depending on the learning conditions you get in the game.

This is not a reflection of your abilities and we are very grateful for your persistence.

You can track your scores throughout the game to see how far you are from reaching your assigned goal.

Please try to find a quiet place to avoid distractions during the session.

Give it a try!

### Briefing

The study is being conducted by Vera Kempe and Nikolay Panayotov, has received ethical approval by the Research Ethics Committee of Abertay University Dundee, Scotland, and complies with the data protection measures specified by GDPR.

It will involve one session and will typically last around 15 to 30 minutes. We ask you to complete the task in one sitting.

Your participation in this study is voluntary.

There are no known risks for you in this study.

You may decide to stop taking part at any time during the study without explanation and without penalty. To do so, simply close your browser at any point. If you do, your data will be destroyed, and will not be included in any analyses or publications.



Any personally identifying information related to this study (e.g. your Prolific ID if applicable) will be kept separately from your responses with an anonymised identifier. Your responses to this study will only be associated with an anonymised identifier. As your data is kept anonymous, we will not be able to remove it after completion of the study. We will not analyse individual data, only the group data. All of the data associated with this study will be stored on password-protected servers to which only the researchers will have access. Your anonymised data will be uploaded to an open, online repository after the completion of testing. However, we will not share your personally-identifiable data on any open repositories.

Researchers are obliged to retain research data for up to 10 years' post-publication, however your anonymised research data may be retained indefinitely (e.g., so that researchers engage in open practice allowing other researchers to access their data to confirm the conclusions from published work). Since your data is stored in an anonymous format, and we do not store your personally identifying information with a recording of your consent to take part, we will be unable to remove your data on completion of the experiment.

If you are interested in the outcome of the study, please email [language.enquiries@abertay.ac.uk](mailto:language.enquiries@abertay.ac.uk).

Please tick the boxes below to provide your consent to take part in our study, and allow for the collection and analysis of your data:

☐ I consent to take part in this study conducted by Vera Kempe and Nikolay Panayotov who intend to use my data for research that examines how people learn and perform in online environments.

☐ I consent for this website to use temporary cookies for strictly essential purposes that are not used for advertisement or other purposes unrelated to the study.

### **The Binary Buzzer Language Game**

**Please have your sound switched ON to continue!**

You will first do a few training rounds to familiarise yourself with the language.

In the first training task you will hear the two-buzzer words for a few strange objects.

Try to work out which object each word refers to.

You will then see the correct object and hear the word again.

If you get it right, you will earn a point. Get enough points in a round and you will receive a star.

When you get two stars your training will be complete.

Don't worry if you feel like you can't really work it out right away - learning a language is not a trivial task. If unsure, just use your intuition. This is not a competition.

**It is important that you do not take any notes!**

**Have fun!**

**Round complete!**

You have X of 2 stars. You will continue the training until you have two.

In the next round of training you will try to remember the words for the objects one by one.

You need to input the sounds of each word with the buttons on the screen.

You get a point if you are right. Get enough points to earn a star.

This task is a bit more difficult, but just try your best.

You are getting there! Keep trying and you will do better on the next round!

**Round complete!**

You have of 2 stars. You will continue the training until you have two.

In this task you will hear the two-buzzer words for the objects.

Try to work out which object each word refers to.

You will then see the correct object and hear the word again.

If you get it right, you will earn a point. Get enough points in a round and you will receive a star.

You are getting there! Keep trying and you will do better on the next round!

**Great Job!**

Now let's see how well you have learned the words.

Try to name all the objects using the low and high buzzes as before.

Make sure it is possible to figure out from your words what objects you mean.

**Excellent!**

Now you can test how clear your words are. Can you understand yourself?

You will hear a few of your own words. Try to identify the meaning.

At the end we will tell you how you did.

Good luck!

**One last thing...**

Before we tell you how well you did, can you please give us some feedback?

How did you find out about the game? \_\_\_\_\_

How much did you enjoy playing the game? (1 - not at all; 5 - very much)

1 2 3 4 5

Please share with us any thoughts you have about the game: \_\_\_\_\_

**Thank you for playing the The Binary Buzzer Language game!**

You got X% right on the last task!

We know this wasn't easy, but you did an excellent job learning! Your contribution is greatly valued.

If you have any questions or feedback, please send us a message  
at: [language.enquiries@abertay.ac.uk](mailto:language.enquiries@abertay.ac.uk)

Want to help science even more? Please share our game on Facebook and Twitter.

## TWO COLOUR LANGUAGE STUDY

### Two Colour Language

#### Can you understand and use a language expressed with only two colours?

Scientists are trying to understand whether humans could communicate using only two colour signals.

See how well you can understand and use such a language! It may take between 15 and 30 min or even a bit more depending on the learning conditions you get in the game.

This is not a reflection of your abilities and we are very grateful for your persistence.

Please try to find a quiet place to avoid distractions during that time.

Thank you for helping our research!

Give it a try!

### Briefing

The study is being conducted by Vera Kempe and Nikolay Panayotov, has received ethical approval by the Research Ethics Committee of Abertay University Dundee, Scotland, and complies with the data protection measures specified by GDPR.

It will involve one session and will typically last around 15 to 30 minutes. We ask you to complete the task in one sitting.

Your participation in this study is voluntary.

There are no known risks for you in this study.

You may decide to stop taking part at any time during the study without explanation and without penalty. To do so, simply close your browser at any point. If you do, your data will be destroyed, and will not be included in any analyses or publications.

Any personally identifying information related to this study (e.g. your Prolific ID if applicable) will be kept separately from your responses with an anonymised identifier. Your responses to this study will only be associated with an anonymised identifier. As your data is kept anonymous, we will not be able to remove it after completion of the study. We will not analyse individual data, only the group data. All of the data associated with this study will be stored on password-protected servers to which only the researchers will have access. Your anonymised data will be uploaded to an open, online repository after the completion of testing. However, we will not share your personally-identifiable data on any open repositories.

Researchers are obliged to retain research data for up to 10 years' post-publication, however your anonymised research data may be retained indefinitely (e.g., so that researchers engage in open practice allowing other researchers to access their data to confirm the conclusions from published work). Since your data is stored in an anonymous format, and we do not store your personally identifying information with a recording of your consent to take part, we will be unable to remove your data on completion of the experiment.

If you are interested in the outcome of the study, please email [language.enquiries@abertay.ac.uk](mailto:language.enquiries@abertay.ac.uk).

Please tick the boxes below to provide your consent to take part in our study, and allow for the collection and analysis of your data:

☐ I consent to take part in this study conducted by Vera Kempe and Nikolay Panayotov who intend to use my data for research that examines how people learn and perform in online environments.

☐ I consent for this website to use temporary cookies for strictly essential purposes that are not used for advertisement or other purposes unrelated to the study.

### **The Two Colour Language Game**

You will first do a few training rounds to familiarise yourself with the language.

In the first training task you will see the two-colour words for a few strange objects.

Try to work out which object each word refers to.

You will then see the correct object and see the word again.

If you get it right, you will earn a point. Get enough points in a round and you will receive a star.

When you get two stars your training will be complete.

Don't worry if you feel like you can't really work it out right away - learning a language is not a trivial task. If unsure, just use your intuition. This is not a competition.

**It is important that you do not take any notes!**

**Have fun!**

### **Round complete!**

You have X of 2 stars. You will continue the training until you have two.

In the next round of training you will try to remember the words for the objects one by one.

You need to input the colours of each word with the buttons on the screen.

You get a point if you are right. Get enough points to earn a star.

Pay attention to the screen while pressing the buttons or you might miss seeing your input.

This task is a bit more difficult, but just try your best.

### **Round complete!**

You have X of 2 stars. You will continue the training until you have two.

In this task you will see the two-colour words for the objects.

Try to work out which object each word refers to.

You will then see the correct object and see the word again.

If you get it right, you will earn a point. Get enough points in a round and you will receive a star.

### **Great Job!**

Now let's see how well you have learned the words.

Try to name all the objects using the two colours as before.

Make sure it is possible to figure out from your words what objects you mean.

Pay attention to the screen while pressing the buttons or you might miss seeing your input.

### **Excellent!**

Now you can test how clear your words are. Can you understand yourself?

You will see a few of your own words. Try to identify the meaning.

At the end we will tell you how you did.

Good luck!

### **One last thing...**

Before we tell you how well you did, can you please give us some feedback?

How did you find out about the game? \_\_\_\_\_

How much did you enjoy playing the game? (1 - not at all; 5 - very much)

1 2 3 4 5

Please share with us any thoughts you have about the game: \_\_\_\_\_

### **Thank you for playing the The Binary Buzzer Language game!**

You got X% right on the last task!

We know this wasn't easy, but you did an excellent job learning! Your contribution is greatly valued.

If you have any questions or feedback, please send us a message

at: [language.enquiries@abertay.ac.uk](mailto:language.enquiries@abertay.ac.uk)

Want to help science even more? Please share our game on Facebook and Twitter.