Mousekids Manual

Martin Schoemann, Anna Lang*, Tilmann Betsch, & Anne Lehmann
Department of Psychology, University of Erfurt, D-99089 Erfurt, Germany

- Corresponding author. E-Mail address: anna.lang@uni-erfurt.de.

Content

1. Introduction

Mousekids is an easily adaptable tool for examining decision making behavior in children and adults. It was designed analogue to the classic “mouselab” paradigm (Payne, Bettman & Johnson, 1988). In contrast to mouselab tools typically used with adults, the Mousekids program contains two different phases: In the learning phase, participants learn the validity of each cue, one after the other; in the subsequent testing phase, participants then make a series of decisions. The cue validities are symbolized throughout the entire game using so-called “smart circles”. Due to the experience-based learning of the cue validities as well as their simple depiction based on absolute frequencies, the Mousekids tool can be used to examine probabilistic inference-based decisions in children even as early as preschool.

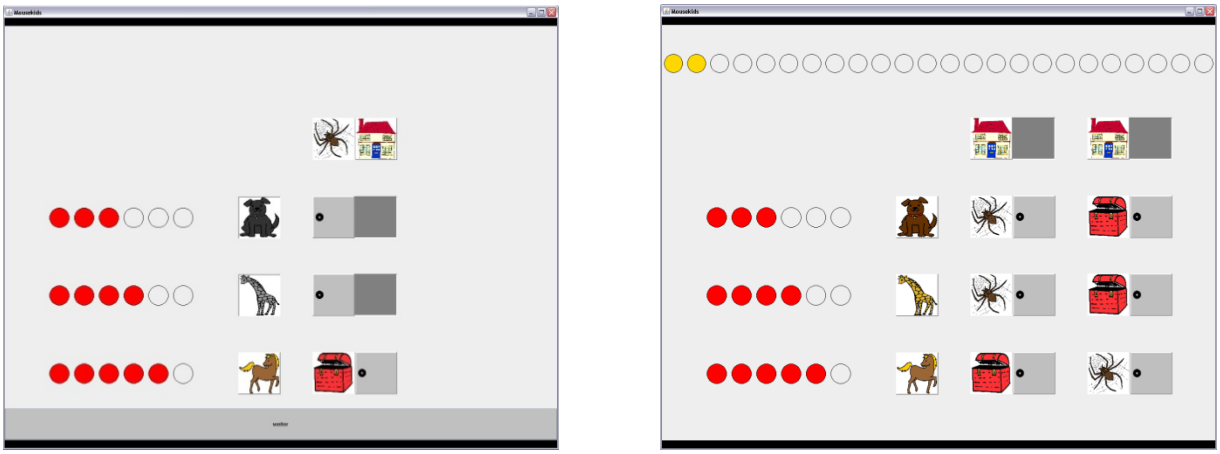


Figure 1: Mousekids in the learning and testing phase. The cue validity is displayed in the form of colored circles next to each cue.

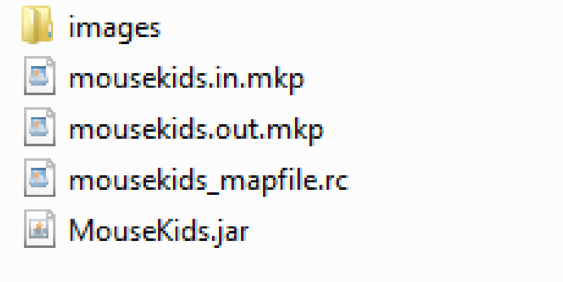
Mousekids was developed so that users can easily adapt the decision and game environments without computer programming skills. For example, one can easily change the number of cues used in the game or the number of options from which participants can choose. In the following sections, you will learn how to modify the Mousekids program to meet your needs.

Mousekids is free of use and can be downloaded from the authors’ homepage at
and used for scientific non-commercial purposes.

2. Procedure during data collection

Before you learn how to modify the Mousekids program, we will first introduce and explain the functions of the program by providing a step by step description as the program runs during data collection. In order to run the program, you will need several data files: the command file (Mousekids.jar); the input file (mousekids.in.mkp), which determines the structure of the game environment; the output file (mousekids.out.mkp), in which the game process and all relevant variables are saved; and the image assignment file (mapfile.rc), which determines which images will appear at which positions in the game.

Prior to beginning data collection, run the command file. The program will ask you if you would like to collect data ("Collect Data") or convert data files ("Merge Data"). Click "Collect Data" and then select the input file that you wish to use (e.g., the file corresponding to the experimental condition that you wish to run). The input file describes how the game environment will appear to participants. Enter the name of the output file that will be created during data collection. The output file name can for example contain the participant number but must always end in ".mkp". Then, select a image assignment file, that describes which images will be used in the game. The Mousekids game will then start automatically. You can then enter the participant number and experimenter number or name and run the game.



images
mousekids.in.mkp
mousekids.out.mkp
mousekids_mapfile.rc
MouseKids.jar

Figure 2: Mousekids files: input file (mousekids.in.mkp), output file (mousekids.out.mkp), image assignment file (mousekids_mapfile.rc), command file (Mousekids.jar) and file folder for images used in the program (images)

3. System requirements

Mousekids will run on any computer with Java Runtime Environment 6 (or newer). The type of operating system is not important, as Java Runtime Environment can be installed on all computers regardless of the platform used. However, due to slight differences in the way in which the game is displayed, we recommend using only computers with the same operating system within a single study. In order to play the game with a touch screen, you of course need correctly installed input devices. If you would like to use a highly complex decision environment, for example by including a large number of options, you will require an adequately large computer monitor.

4. Modifying the game environment

Mousekids allows for numerous possibilities to modify the game environment. Using the input file, you can, for example, change the number of cues and options to adapt the decision situation to your needs. In the following table, you will find an overview of the available modification options.

	Variable	**Input file name**
General	Number of cues	RowCount
	Number of options	ColumnCount
	Images for cues	Determined in image assignment file
	Images for doors and options	
	Images for hits and misses	
	Touch screen	TouchScreen (yes/ no)
Learning phase	Number of learning trials per cue	NumberOfLearningBoards
	Cue predictions and outcomes in the learning phase	LEARNINGBOARDS
	Number of learning phases	NumberOfLearningPhases
Testing phase	Number of decisions	NumberOfDecisionBoards
	Number of practicing trials to learn the game	NumberOfTrials
	Time point of play breaks	BreakAfterDB
	Cue predictions and outcomes in the testing phase	DECISIONBOARDS
	Presentation format for the cues‘ predictions	VisibilityFactor (open/ closed/ closing/ closingClick)
	Feedback about decision outcomes	ImmediateFeedbackFactor (yes/ no/ complete)
	Position of cues for each decision	DecisionBoardRowOrder
	Order of decisions	DecisionBoardOrder

Table 1: Modification possibilities in the Mousekids program

You can modify the game environment using the input file. This file determines how the learning and testing phase will appear. If you would like to create different game environments, you will need a separate input file for each game environment.

To modify the game environment, open the input file in a text editor program. We recommend that you do not use Microsoft Editor. Instead, we suggest using a program that offers better usability, such as WordPad. In the following section, we will describe in detail how to modify each of the variables displayed in the table above.

4. 1. Images

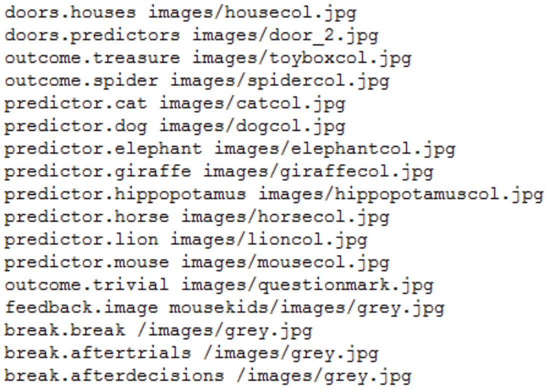
The images used to depict the cues and options and all other pictures are determined in the image assignment file. You should save all image files that will be used by Mousekids to a single file folder that is located in the same folder as the image assignment file. All images must be saved in the jpeg format.

We will describe how to change the images displayed in the game using the following example. Imagine that you would like to modify the game so that participants have to choose between cars rather than houses:

Open the image assignment file. In the first line you will see the text “doors.houses images/housecol.jpg“. This command tells the program that Mousekids should use the image “housecol.jpg” in the folder “images” to display the options (“doors.houses”).

If you want the game to display the image of a car instead, you simply change the command “images/housecol.jpg“ to, for example, “images/car.jpg“. The image car.jpg must be located in the folder “images”.

In the same manner, you can also change the images depicting the cues, outcomes and doors.



```
doors.houses images/housecol.jpg
doors.predictors images/door_2.jpg
outcome.treasure images/toyboxcol.jpg
outcome.spider images/spidercol.jpg
predictor.cat images/catcol.jpg
predictor.dog images/dogcol.jpg
predictor.elephant images/elephantcol.jpg
predictor.giraffe images/giraffecol.jpg
predictor.hippopotamus images/hippopotamuscol.jpg
predictor.horse images/horsecol.jpg
predictor.lion images/lioncol.jpg
predictor.mouse images/mousecol.jpg
outcome.trivial images/questionmark.jpg
feedback.image mousekids/images/grey.jpg
break.break /images/grey.jpg
break.aftertrials /images/grey.jpg
break.afterdecisions /images/grey.jpg
```

Figure 3: The image assignmen file

The following table provides an overview of the images that can be changed as well as their command codes in the image assignment file.

Images	Command code in image assignment file
Image for options	doors.houses
Image for doors obscuring the cues’ predictions	doors.predictor
Image for positive outcome (hit)	outcome.treasure
Image for negative outcome (miss)	outcome.spider
Image for outcome when no feedback is provided	outcome.trivial
Image for cue „cat“	predictor.cat
Image for cue „dog“	predictor.dog
Image for cue „n“	predictor.n
Image after practicing trials	break.aftertrials
Image in play breaks	break.break
Image after testing phase	break.afterdecisions
Image at the end of the game	feedback.image

Table 2: Descriptions and command code of user-defined images

4. 2. Number of cues and options

You can change the number of cues and options by entering the desired number of cues under „RowCount“ and the desired number of options under „ColumnCount“ in the input file. You can then enter the names of the cues under “Predictors” and define the images that will be used for each cue in the image assignment file. If you would like to change the number of cues or options, you must modify the prediction patterns in the learning and testing phase as well as the order of the cues to fit the desired number of cues and options.

4.3. Touch screen

In order to ensure that the game functions properly, enter whether participants will be playing the game using a touch screen (yes / no) under „Touchscreen“.

4.4. Number of learning trials per cue and number of learning phases

Under „NumberofLearningBoards“ you can define the number of trials that will be used to learn each cue validity. You can then define the predictions of the cues and outcomes under “[LEARNINGBOARDS]” (see section 4.5). It is also possible to increase the number of learning phases and allow participants to complete the whole learning phase multiple times. To do so, simply enter the desired number of learning phases under “NumberofLearningPhases”. It is possible to set the number of learning phases to 0. However, this means that no cue validities will be displayed during the game. If you set the number of learning phases to 0, you should also set the variable “NumberofLearningBoards” to 0.

4.5. Learning phase patterns

You can define the predictions of the cues and outcomes and, with that, the cue validity of each cue. Negative predictions and outcomes are entered as “s” (for spider). Positive predictions and outcomes are entered as “t” (for treasure). These prediction-outcome patterns are organized in blocks in the input file. The first block always contains the predictions of the first, second, *n*th cues for the first learning trial and the respective outcomes in the first line.

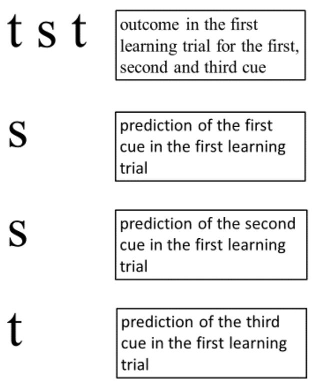

t s t
outcome in the first learning trial for the first, second and third cue
s
prediction of the first cue in the first learning trial
s
prediction of the second cue in the first learning trial
t
prediction of the third cue in the first learning trial

Figure 4: Learning phase pattern: The block contains all predictions of the first, second, third cue in the first learning trial as well as the corresponding outcomes.

4.6 Number of decisions

The number of decisions in the testing phase can be defined under „NumberOfDecisionBoards“. The outcome-prediction patterns have to be configured for each decision separately.

4.7. Testing phase patterns

The predictions of the cues and outcomes for the different options are defined under „[DECISIONBOARDS]“. Here, negative predictions and outcomes are again defined by “s” (for spider), whereas positive predictions and outcomes are defined by “t” (for treasure). The patterns are organized in blocks, where the first block describes the pattern of the first decision. The first line defines the outcomes for all the options. The second, third, and fourth lines define the predictions of the first, second, and third cues for the respective options.

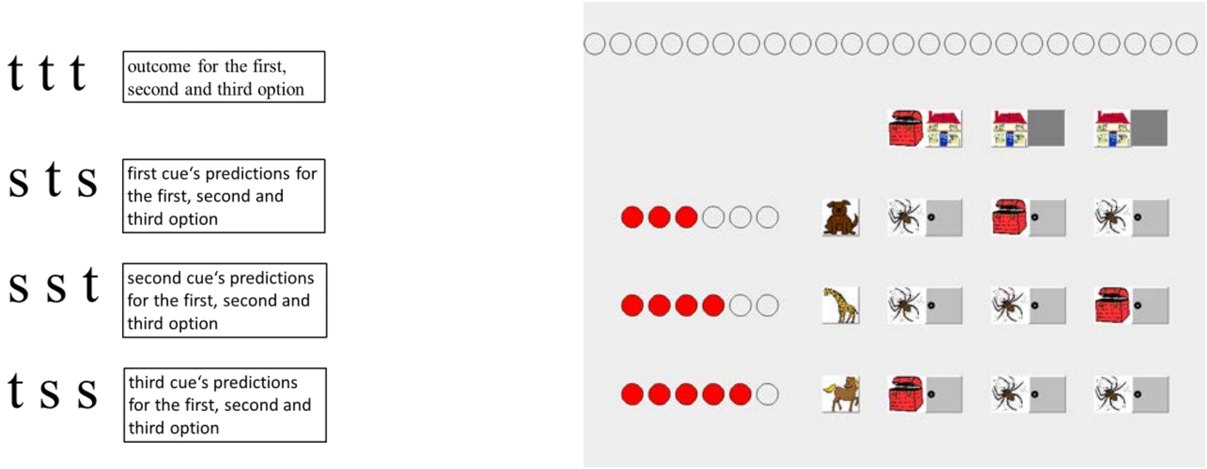

t t t
outcome for the first, second and third option
s t s
first cue‘s predictions for the first, second and third option
s s t
second cue‘s predictions for the first, second and third option
t s s
third cue‘s predictions for the first, second and third option

Figure 4: Testing phase pattern in the input file and in the game

Although the position of the cues may change for each decision (see section 4.12.), this does not have an effect on the description of the pattern. The predictions of the cue whose validity was learned first in the learning session are always displayed in the second line. The predictions of the cue whose validities were learned second in the learning phase are always displayed in the third line, and so forth.

Please note that you may also define the first decisions in the testing phase as practicing trials (see section 4.8).

4.8. Number of practicing trials

In order to provide participants with the opportunity to learn the game, you can define an arbitrary number of practicing trials (however, a maximum of “NumberofDecisionBoards“-1). During the practicing trials, participants can select an option to display its outcome but are not awarded points for correct answers. If you do not want to use practicing trials, enter 0 under “NumberofTrials”.

4.9. Play breaks

You can define an arbitrary number of play breaks by entering the decision after which the break should take place under „BreakAfterDB“. For instance, when you want to have a play break after every third decision, enter “3 6 9” and so forth. During the game, after every third decision a “continue” button will then be displayed rather than the next decision. If you do not want to have play breaks, do not enter a number after “BreakAfterDB”.

4.10. Visbility

In the testing phase different presentation formats can be used to display the cues’ predictions. In an open presentation format all pieces of information can be inspected simultaneously. In a closed presentation format all pieces of information are hidden and have to be uncovered first. After being uncovered once all pieces of information are visible until the participant makes her decision. In the “closingClick” presentation format doors close after the next prediction is inspected so that only one piece of information is visible at a time. In the “closing” presentation format each door closes after a predefined period of time, which must be specified under “ClosingTime” (in miliseconds). During that time period participants cannot open another door.

4.11. Feedback

Mousekids can be played either with, without or with complete feedback. With feedback means that participants are informed after each decision about the outcome of the chosen option. Without feedback means that participants receive no information about the outcome of their decision. When participants receive complete feedback they are informed of the outcomes of all options, the chosen and the non-chosen one(s).

However, even without feedback, the number of correct decisions made during the game is tracked and displayed at the end of the game to allow for rewarding participants based on their performance. If feedback is used, a point must be awarded after every decision before one can move on to the next decision. In order to keep all game formats as similar as possible, participants not receiving feedback also select and open an option. For this type of outcome you can define a separate image (e.g., a question mark, see table 2 “outcome.trivial). Without feedback participants of course cannot be rewarded treasure points during the game. Instead we implemented “counting points” to keep all game formats as similar as possible. For every decision made a “counting point” has to be clicked on in order to get to the next decision.

In the complete feedback condition participants are informed about the outcomes of all options. After having chosen and opened an option the outcome of the other option(s) must be inspected in order to get to the next decision.

Regardless of whether the game is run with, without or with complete feedback, the first decisions of the testing session can be defined as a practicing trials in which the outcome of the chosen option are displayed but no treasure points are awarded.
You can define which kind of feedback should be given under “ImmediateFeedbackFactor”.

4.12. Position of cues for each decision

The position of the cues can be defined separately for each decision under “DecisionBoardRowOrder”. The number “1” always indicates the cue whose validity was learned first in the learning phase – i.e. the cue corresponding to the predictions displayed in the second line of the pattern block (see section 4.8). Please note that even if you want to order the cues in the same manner as in the learning phase, you still must define the order for each individual decision (in this case “1 2 3”). Here, one line represents one decision.

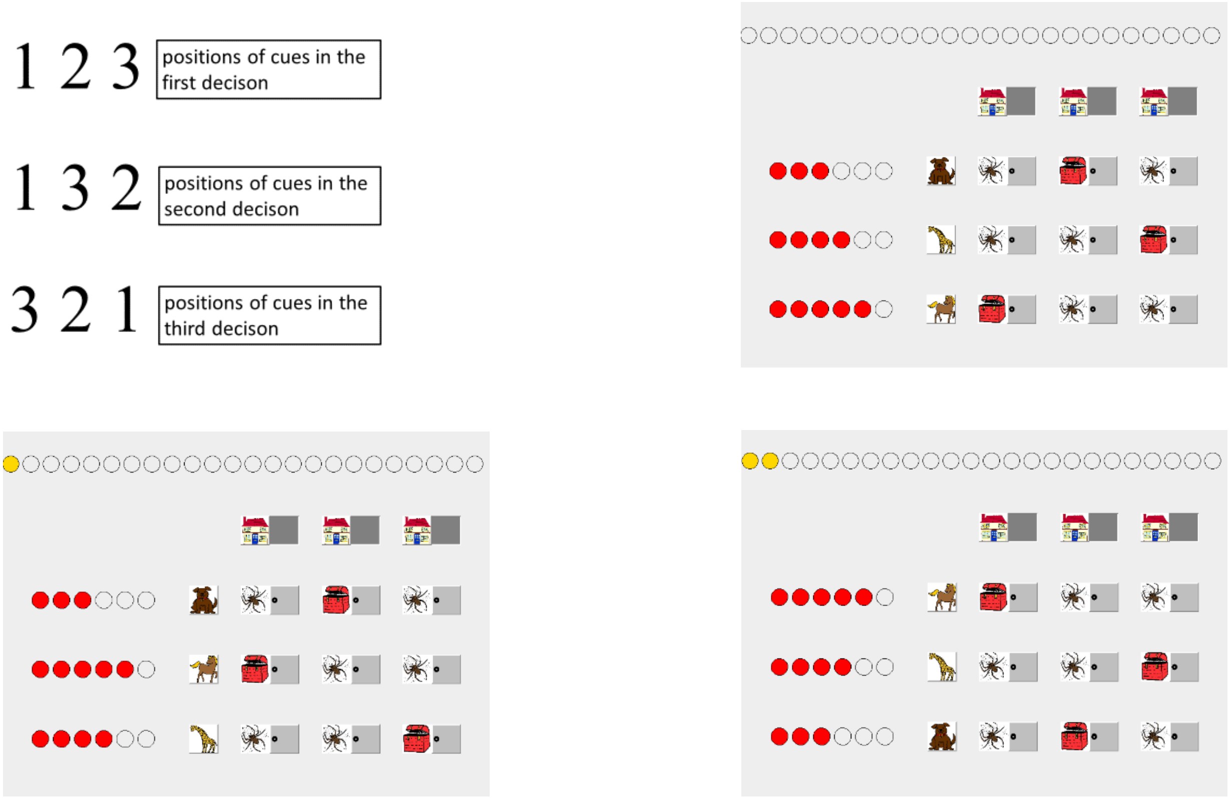

1 2 3
positions of cues in the first decison
1 3 2
positions of cues in the second decison
3 2 1
positions of cues in the third decison

Figure 5: Positions of cues in the first, second and third decision in the input file and in the game

4.13. Including an Instruction Manual

Mousekids was designed to be conducted by an experimenter. However, it is possible to use an instruction manual instead. If you want to include an instruction manual the program tells the participants at several points of the game to go back to the instruction manual (after each cue validity is learned in the learning phase). It is also possible to include instructions directly in the game: after the practicing trials, in the breaks and after the last decision in the testing phase. In order to do that you have to save instructions as pictures and include them using the image assignment file (see table 2). If you want to use an instruction manual enter “yes” under “InstructionManual”.

5. Saving and exporting data

Mousekids saves all data: the number and order of the searched information as well as the decision time and selected option for each decision. The data for each participant are saved in an individual output file. The individual data files can then be combined into a single Excel file.

5.1. Combining datasets

Save all output files that you want to combine to the same folder. Make sure that all output files are in the same format – i.e., have the same number of cues and options.

- Start Mousekids and select „Merge Data“
- Select the folder containing the individual datasets
- Enter the name you want for the Excel file and add the file ending “.xlsx”
- If the files were successfully merged, the notice “All Mousekids protocols are processed successfully.” will appear. If this notice does not appear, one or more of the individual files contained an error. This is most often the case when a participant number is missing or was entered in the wrong format or when one or more of the output files does not contain any data.

5.2. Saved variables

The resulting Excel file contains variables that describe the participants’ search and decision behavior. Search behavior is of course only recorded if the cues’ predictions are hidden. The following table provides an overview of the variable names and codes contains in the data files.

Variable	**Name in Excel file**	**Coding**
Selected option in the first, second and nth decision	D1_Choice D2_Choice Dn_Choice	1 , 2, 3, …, n Options are numbered from left to right.
Information that was viewed first in the first decision	D1_Door1	11, 12, 13 The position of the searched information is described as a combination of two numbers in which the first number refers to the column and the second number to the row (e.g., 23 = second column, third row). The number of doors that are saved is limited to n_{max} = number of doors in the matrix*3. Missings are defined as -66.
Information that was viewed second in the first decision	D1_Door2	
Information that was viewed n^{th} in the first decision	D1_Doorn	
Number of opened doors in the first decision	D1_open	1, 2, 3,… n (n_{max} = number of doors in the matrix)
Number of times a door was opened	D1_open_freq	1, 2, 3,… n
Doors opened multiple times	D1_open _twice	0 (no), 1 (yes)
Frequency of doors that were opened multiple times	D1_open_twice_freq	1, 2, 3,… n
Decision time for the first decision in miliseconds	D1_time	

Table 3: Names and descriptions of the variables describing search and decision making behavior

The Excel file also includes additional variables describing the game environment seen by the respective participant. This allows you to see which experimental condition the participant was in as well as use this information in statistical analyses. The following table provides an overview of the variable names and codes.

Variable	**Name in Excel file**	**Coding**
Participant number	VP	Number
Presentation format for the cues‘ predictions in the testing phase	VisibilityFactor	open, closed, closing, closingClick (string)
Feedback after every decision	ImmediateFeedbackFactor	yes,no, complete (string)
Number of learning phases	NumberOfLearningPhases	0,1,…n
Number of practicing trials	NumberOfTrails	0,1,…n
Instruction Manual	InstructionManual	0
Cue whose validity was learned first in the learning phase	Cue1	0,1, …, n Cues are coded according to their order in the input file (e.g input file: cat, dog, mouse; 1= cat, 2= dog,…)
Cue whose validity was learned second in the learning phase	Cue2	
Cue whose validity was learned nth in the learning phase	Cuen	
Prediction of the first cue for the first option for the first decision	D1_Cue1_Prediction1	0, 1
Prediction of the first cue for the second option	D1_Cue1__Prediction2	
Prediction of the first cue for the nth decision	D1_Cue1__Prediction N	
Prediction of the nth cue for the first option	D2_CueN_Predicition1	
Prediction of the nth cue for the second option	D2_CueN_Predicition2	

Variable	**Name in Excel file**	**Coding**
Prediction of the nth cue for the nth option	D2_CueN_Predicition N	0,1
Outcome of the first option in the first decision	D1_Outcome1	
Outcome of the second option in the first decision	D1_Outcome2	
Outcome of the nth option in the first decision	D1_Outcome N	
Prediction of the first cue in the first learning trial	L1_Cue1_Predicition	
Prediction of the second cue in the first learning trial	L1_Cue2_Prediction	
Prediction of the nth cue in the first learning trial	L1_Cue N_Prediction	

Table 4: Names and descriptions of the variables describing the game environment

More variables, e.g. the name or number of the experimenter, the number of gained treasure points are saved in each output file but not converted into the Excel file.

References:
Payne, J. W., Bettman, J. R., & Johnson, E. J. (1988). Adaptive strategy selection in decision making. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 14*, 534–552.