



# **CREATING MOUSE-TRACKING EXPERIMENTS AND ANALYZING MOUSE-TRACKING DATA**

Pascal Kieslich (University of Mannheim) & Dirk Wulff (University of Basel)  
Workshop at the EADM Summer School 2018 in Salzburg, Austria

# Workshop agenda

1

## Mouse-tracking introduction (Monday)

- General introduction
- Your task
- Develop & present experimental design

## Creating mouse-tracking experiments (Tuesday)

- Introduction to OpenSesame & mousetrap-os plugin
- Build & preregister experiment
- Run experiments

## Analyzing mouse-tracking data (Wednesday)

- Introduction to R & mousetrap package
- Covering both basic and advanced analyses and visualizations
- Analyze your data

# Preparations (before the workshop)

2

- Read book chapter by Kieslich et al. (in press)
- Outline two example experiments in your group (meeting the outlined requirements) and describe them in a paragraph
- Upload your ideas in one file name 'GroupX.doc' onto OSF (Project Ideas)



# **DAY 1:**

# **MOUSE-TRACKING INTRODUCTION**

Pascal Kieslich (University of Mannheim)

Workshop at the EADM Summer School 2018 in Salzburg, Austria

# Mouse-tracking introduction (Monday)

4

- 13:00-14:30 General introduction to mouse-tracking
  - ▣ Paradigm and assumptions
  - ▣ Implementation and analysis
  - ▣ Previous applications
- 14:30-15:00 Introduction to task
  - ▣ Type of experiments considered
  - ▣ Your tasks during the workshop
- 15:00-17:00 Develop experimental design conceptually
- 17:00-18:00 Present experimental design in plenum



## Paradigm & assumptions

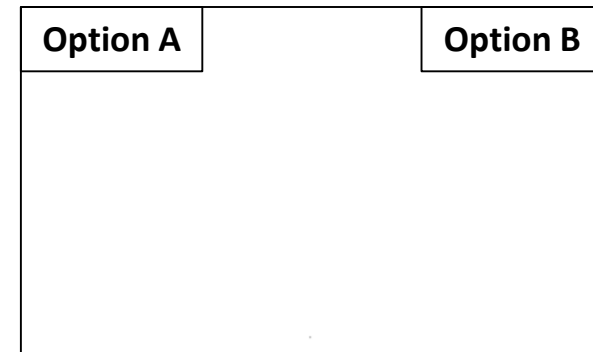
# Mouse-tracking

## Paradigm & assumptions

6

### □ Mouse-tracking (aka. response dynamics)

- ▣ Continuous recording of mouse movements
- ▣ while participants decide between different spatially separated options on a screen



### □ Assumptions

- ▣ Cognitive processing continuously revealed in motor responses (Spivey & Dale, 2006)
- ▣ “Hand in motion reveals mind in motion” (Freeman et al., 2011)
- Mouse movements reveal tentative commitments to and conflict between choice options during decision process

# Mouse-tracking

Seminal article by Spivey et al. (2005)

7

## □ Study on **spoken word recognition**

▣ Instruction: “Click the **candle**”

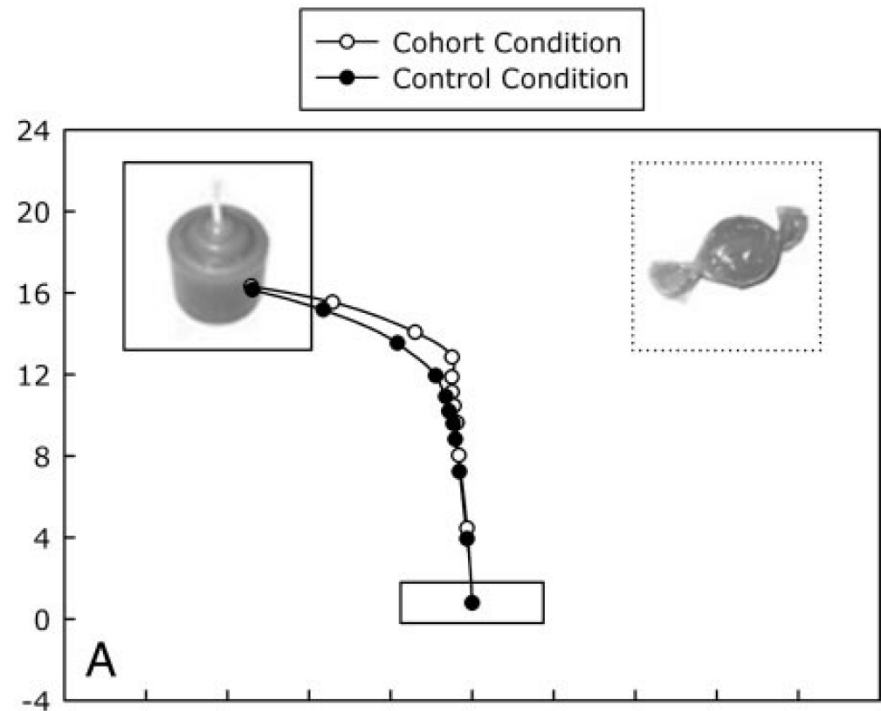
## □ Spatial attraction of hand movement

▣ Greater towards phonologically similar distractor (“**candy**”)

▣ Than towards phonologically dissimilar distractor (“**dice**”)

## □ Evidence

▣ Suggests parallel processing of auditory input activating competing representations





# Mouse-tracking

## Main applications

8

- Mouse-tracking allows for **testing psychological theories**
- Two major applications (cf. review by Stillman et al., 2018)
  - ▣ Provides fine-grained measure for **amount of conflict** between response options
    - test predictions about which factors (contextual factors, individual differences) influence amount of conflict for specific decision
  - ▣ Assess **temporal development** of conflict and its resolution
    - test models that make predictions how decisions unfold over time (e.g., decide between single vs. dual process models)

# Mouse-tracking

## Application domains

9

- Application of mouse-tracking in a **growing number of psychological domains** (Reviews by Freeman, in press; Stillman et al., 2018)
  - ▣ Semantic processing (e.g., Spivey et al., 2005; Dale & Duran, 2011)
  - ▣ Social cognition (e.g., Freeman et al., 2008; Freeman & Ambady, 2011)
  - ▣ Learning and memory (e.g., Dale et al., 2008; Koop & Criss, 2016)
  - ▣ Self-control (e.g., Sullivan et al., 2015; Stillman et al., 2017)
  
- In the last years also extended to **JDM research**
  - ▣ Intertemporal choice (Dshemuchadse et al., 2013)
  - ▣ Moral dilemmas (Koop, 2013)
  - ▣ Decisions under risk (Koop & Johnson, 2013)
  - ▣ Social dilemmas (Kieslich & Hilbig, 2014)
  - ▣ Judgmental biases (Szasz et al., 2018; Travers et al., 2016)



# Implementation & analysis

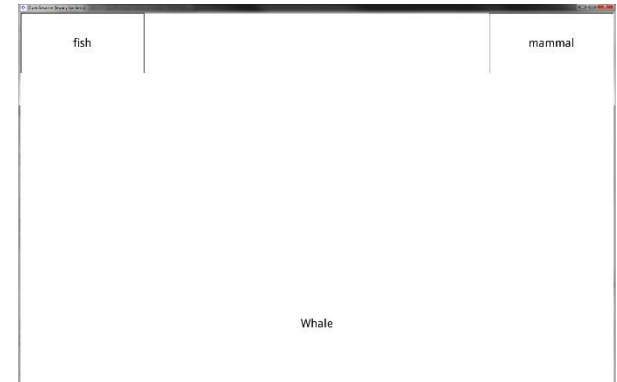
# Implementation & analysis

## Replication study of Dale et al. (2007)

11

### □ Animal categorization task

- ▣ **Typical exemplars** only share features with correct category (e.g., cat as mammal)
- ▣ **Atypical exemplars** share both features with correct and competing category (e.g., whale with mammal and fish)



### □ Main hypothesis

- ▣ **Increased competition** when categorizing atypical exemplars
  - Mouse trajectories with deviation towards competing category

### □ Replication study (Kieslich & Henninger, 2017)

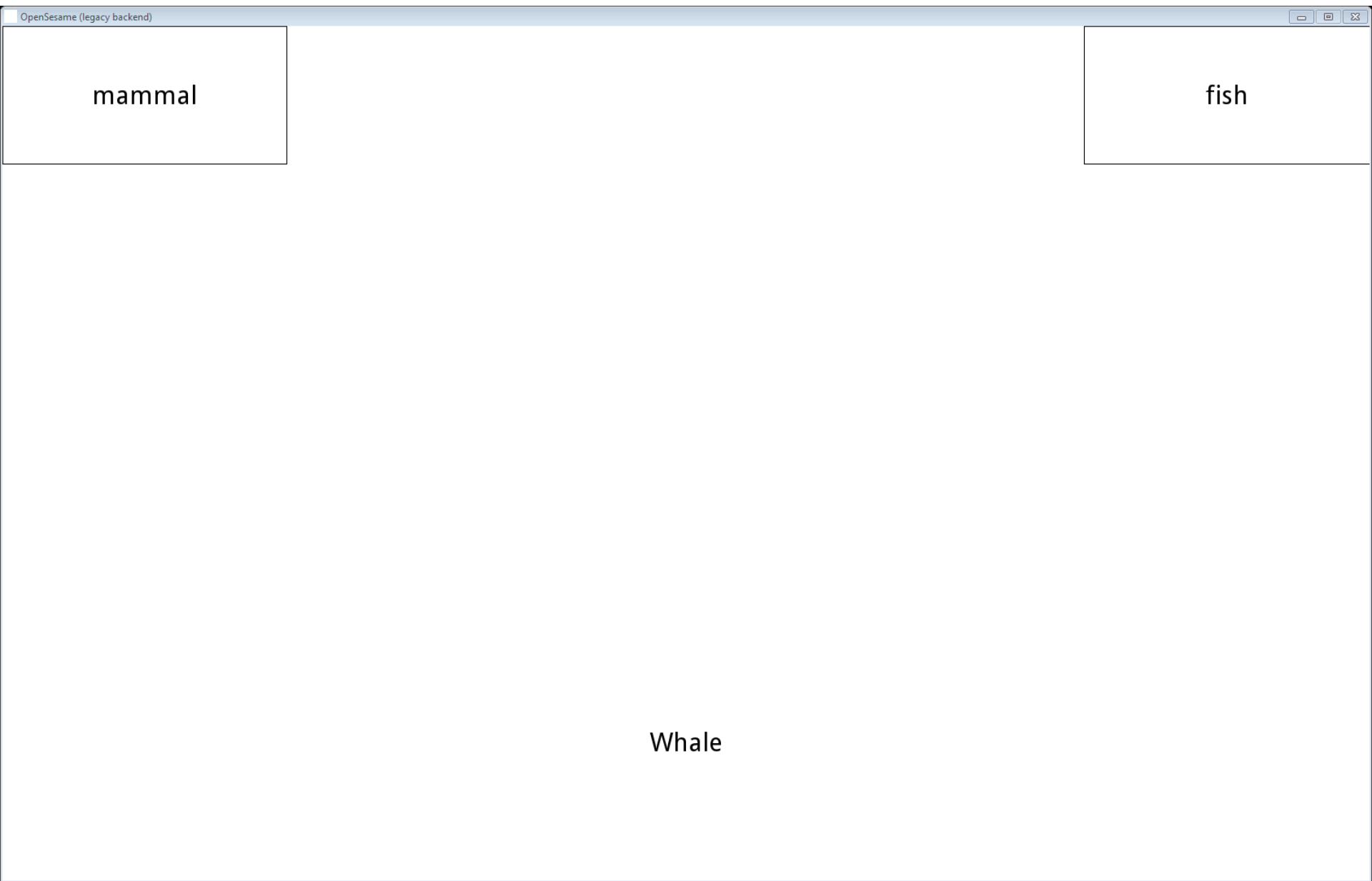
- ▣ Same material (translated into German) and procedure, but higher resolution and different aspect ratio
- ▣  $N = 60$  students from the University of Mannheim
- ▣ Material, data, and analyses at <https://github.com/pascalkieslich/mousetrap-resources>

# Implementation & analysis

## Methodological considerations

12

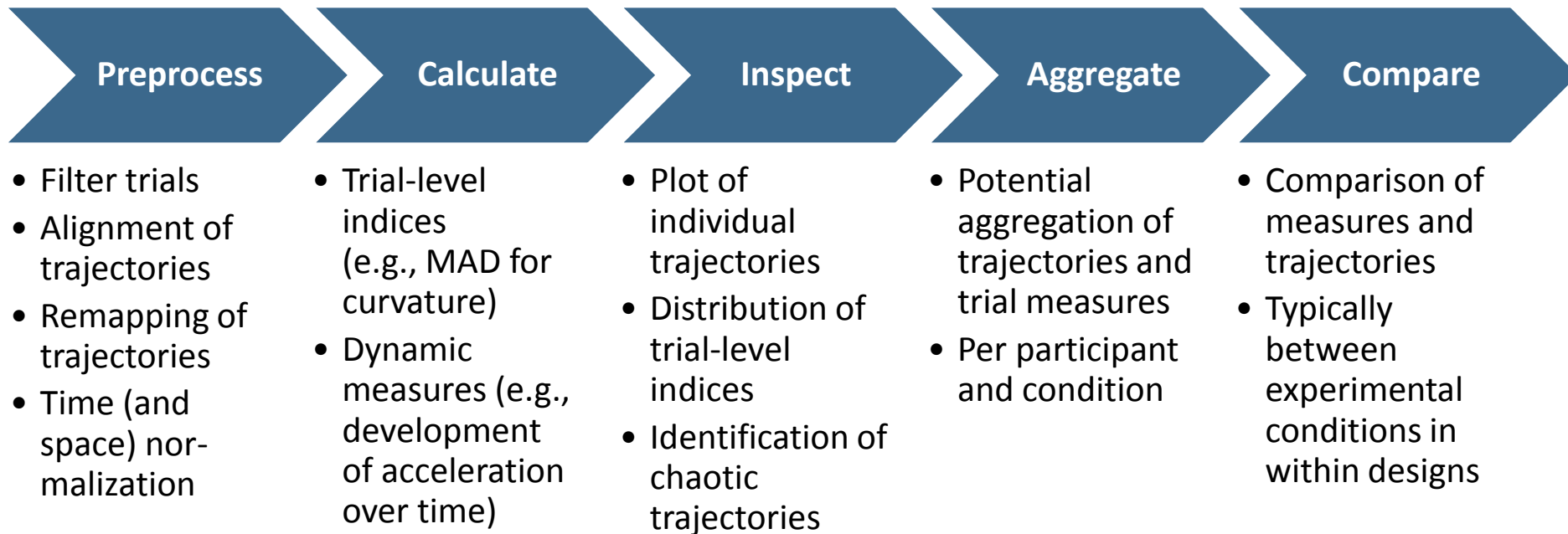
- General challenge when designing a mouse-tracking study
  - ▣ Movements should reflect developing **commitment** not information search (≠ eye-tracking or Mouselab)  
→ **minimize** amount of **new information** after tracking onset
  - ▣ **Preferences** should not develop **before** tracking starts  
→ **critical information** should only be made available at the **last** moment
- Mouse **start positions** should be **comparable** across trials
  - ▣ Participants have to click on a **centered button** to start the trial
  - ▣ Exactly identical start positions across trials achieved by **resetting** mouse or by **computational alignment** during analysis
- **Counterbalancing positions** across trials / participants
  - ▣ Vary which option is presented on which side (left vs. right)
  - ▣ Can be done between trials or between participants (depending on study)



# Implementation & analysis

## Typical analyses steps

14



Analyses steps implemented in the mousetrap R package

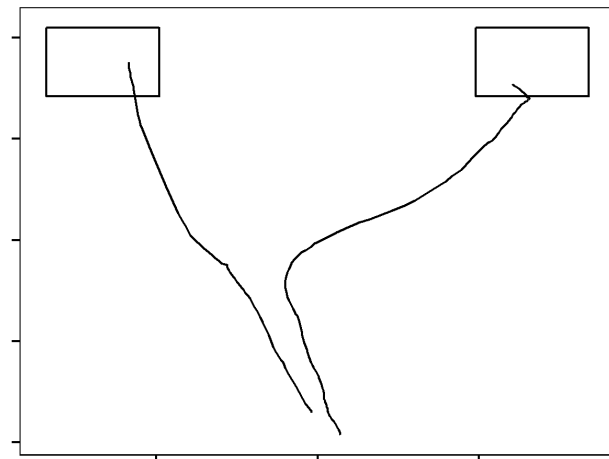
More information: <http://pascalkieslich.github.io/mousetrap/>

Available from CRAN: `install.packages("mousetrap")`

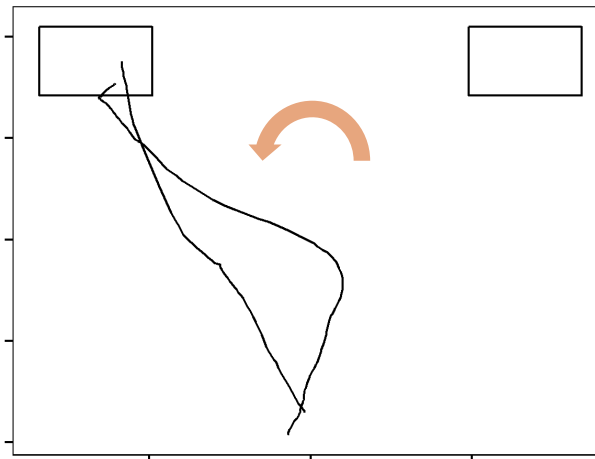
# Implementation & analysis

## Data preparation: Remapping and alignment

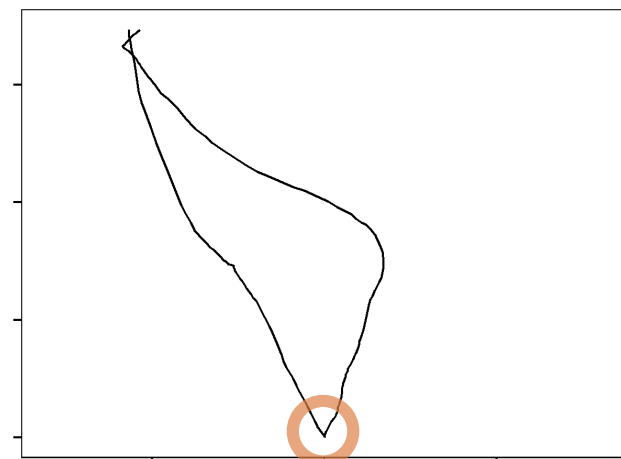
15



**Raw data**



**Remapping**  
Equal direction



**Alignment**  
Equal starting point



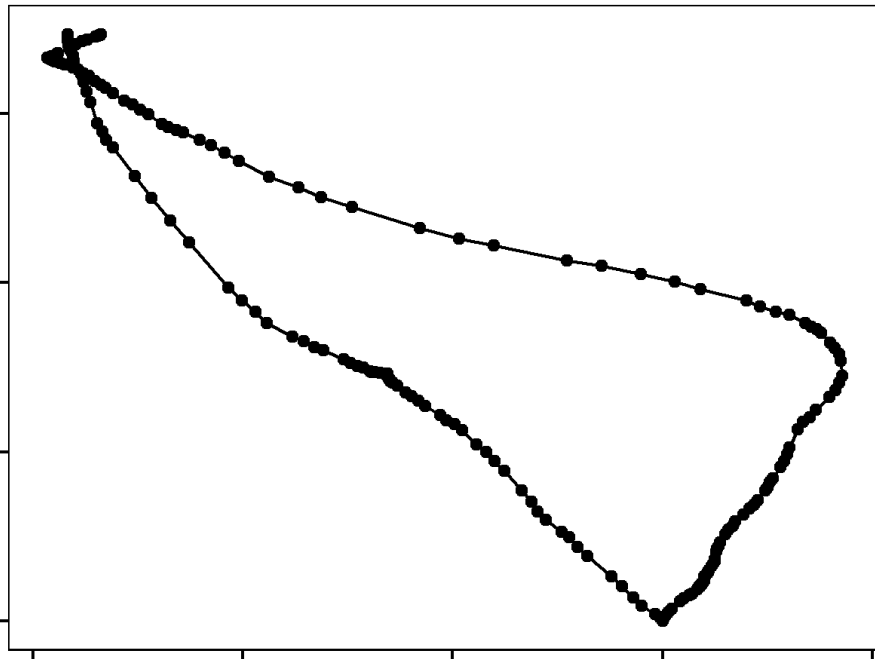


# Implementation & analysis

## Data preparation: Time normalization

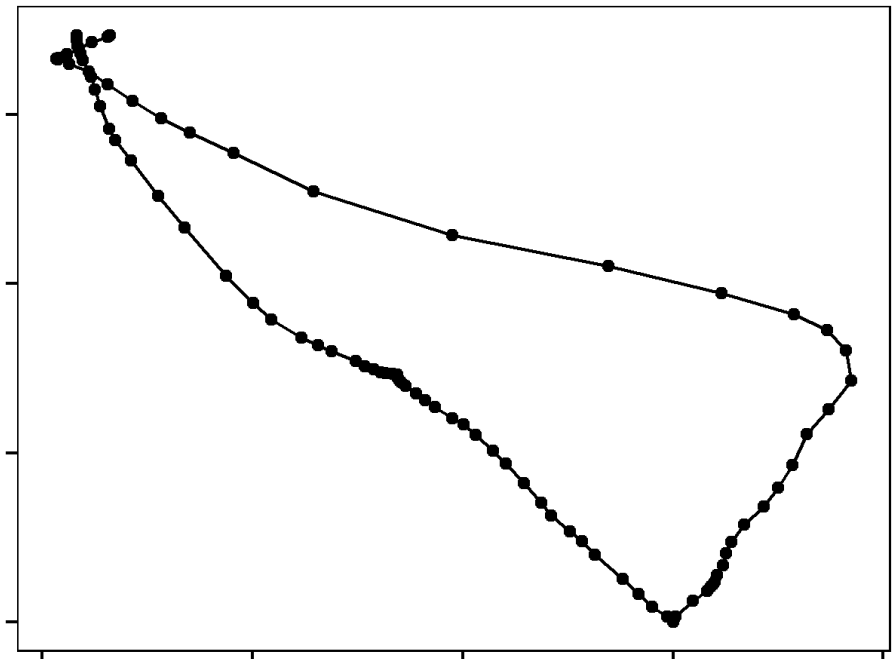
17

- Trials with differing response time vary regarding number of recorded coordinates
- To permit averaging across trials: time-normalization (cf. Spivey et al., 2005)
- Each trajectory divided into 101 equally spaced time steps using linear interpolation



**Raw data**

Constant sampling rate → Absolute time



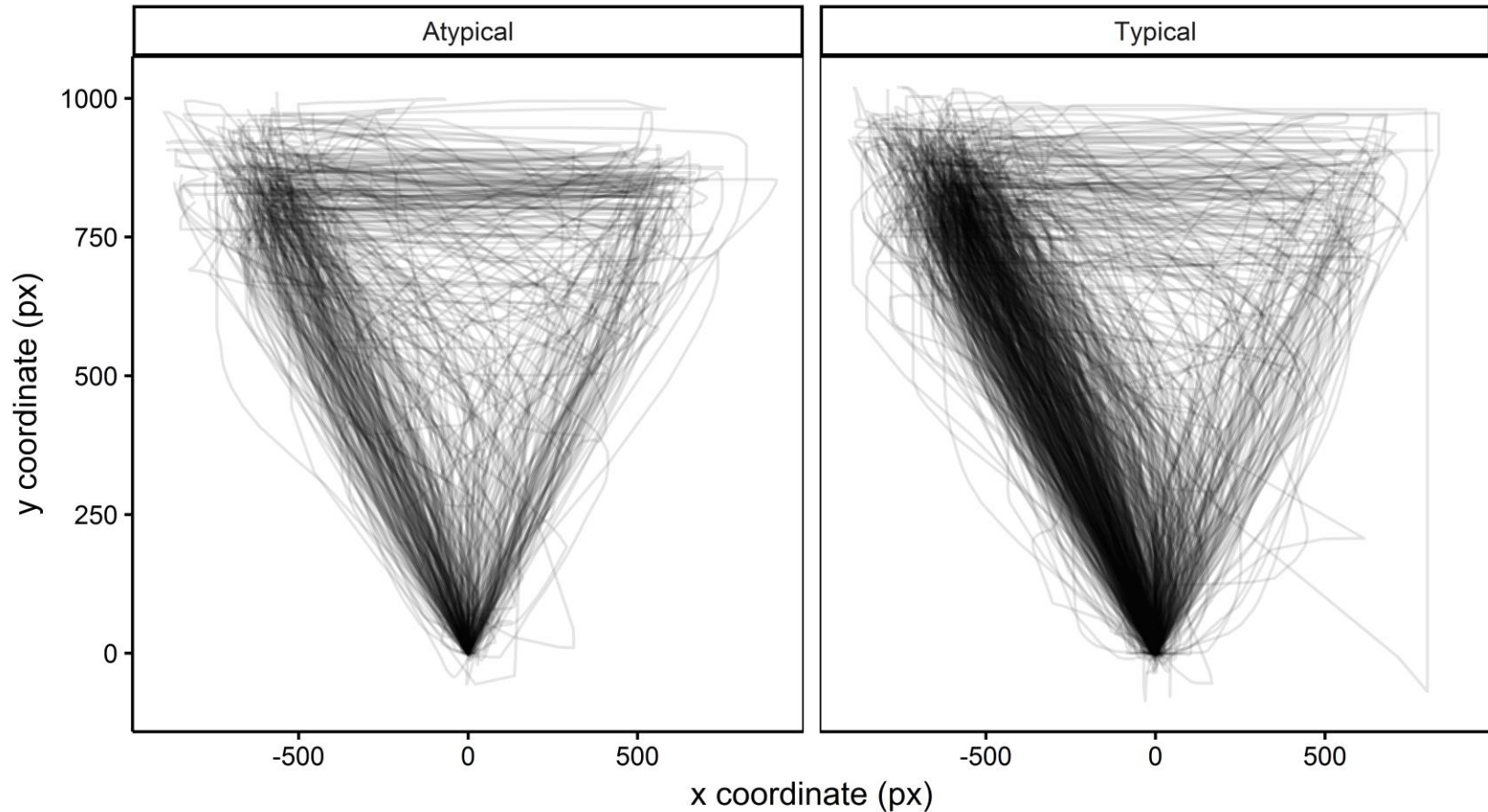
**Time normalization**

Relative time steps

# Implementation & analysis

## Time-normalized trajectories per condition

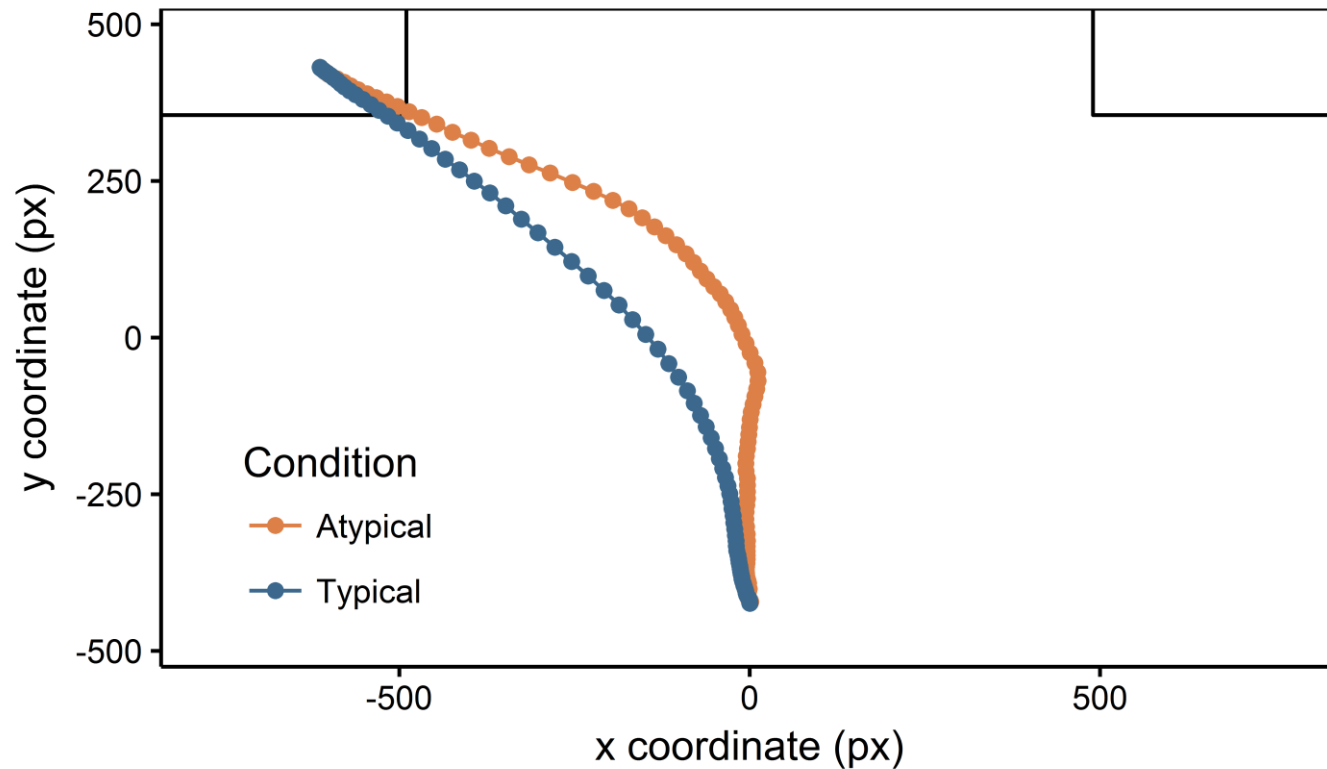
18



# Implementation & analysis

## Average time-normalized trajectories

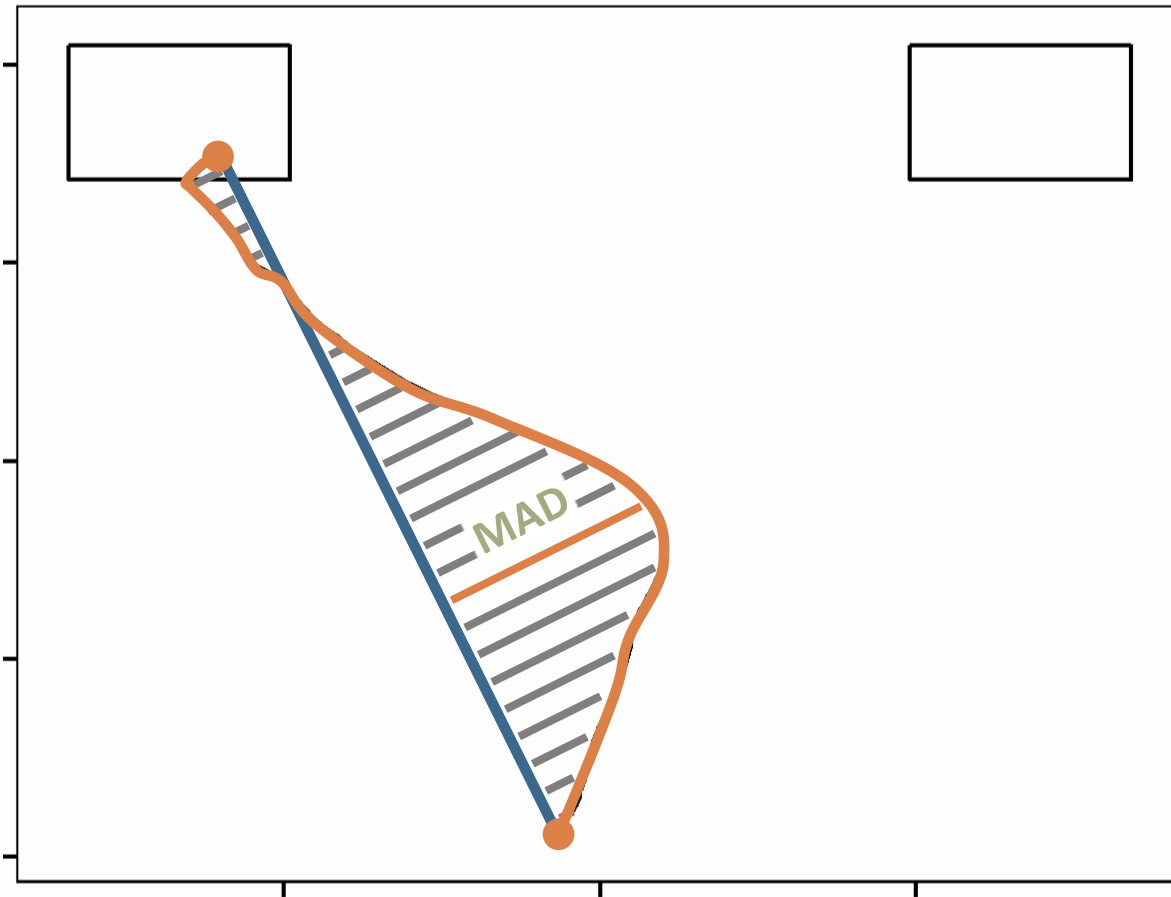
19



# Implementation & analysis

## Selected measures for trajectory curvature

20



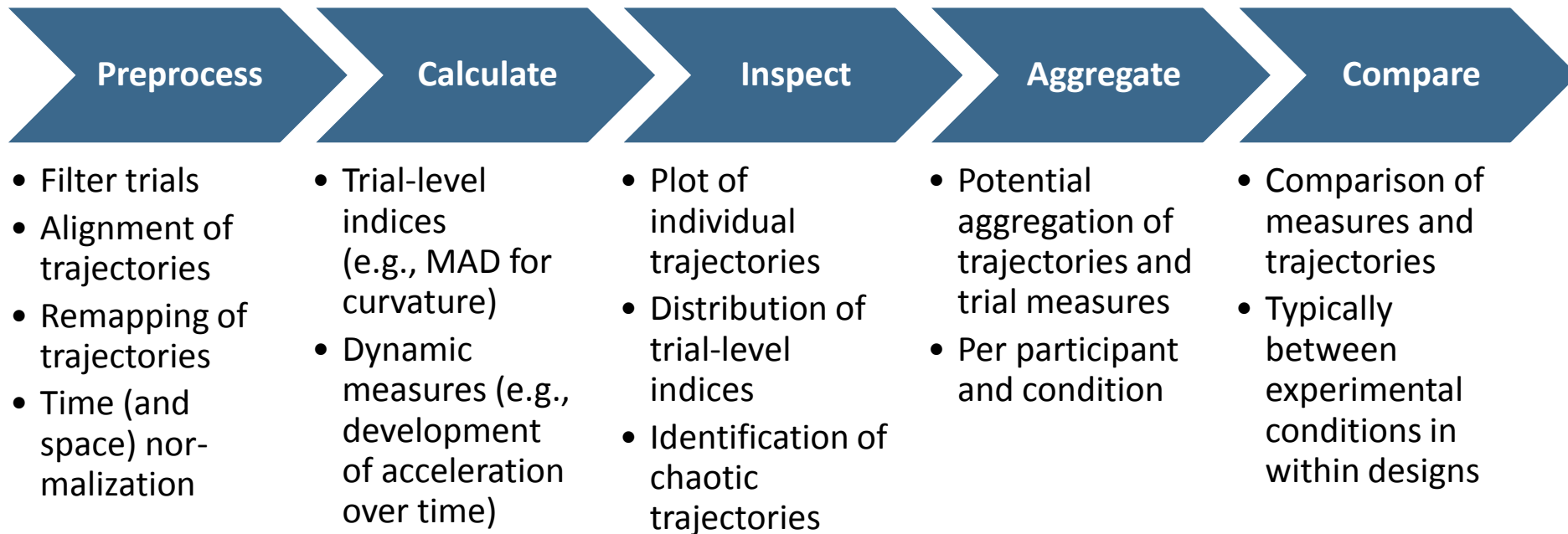
Measures of curvature quantify perpendicular distance between **observed trajectory** and an **idealized straight line**

- Maximum absolute deviation (**MAD**)  
McKinstry, Dale, & Spivey (2008)
- Average deviation (**AD**)  
Koop & Johnson (2011)
- Area under curve (**AUC**)  
Spivey, Grosjean, & Knoblich (2005)

# Implementation & analysis

## Typical analyses steps

21



Analyses steps implemented in the mousetrap R package

More information: <http://pascalkieslich.github.io/mousetrap/>

Available from CRAN: `install.packages("mousetrap")`

# Implementation & analysis

## Comparison of (maximum) absolute deviations

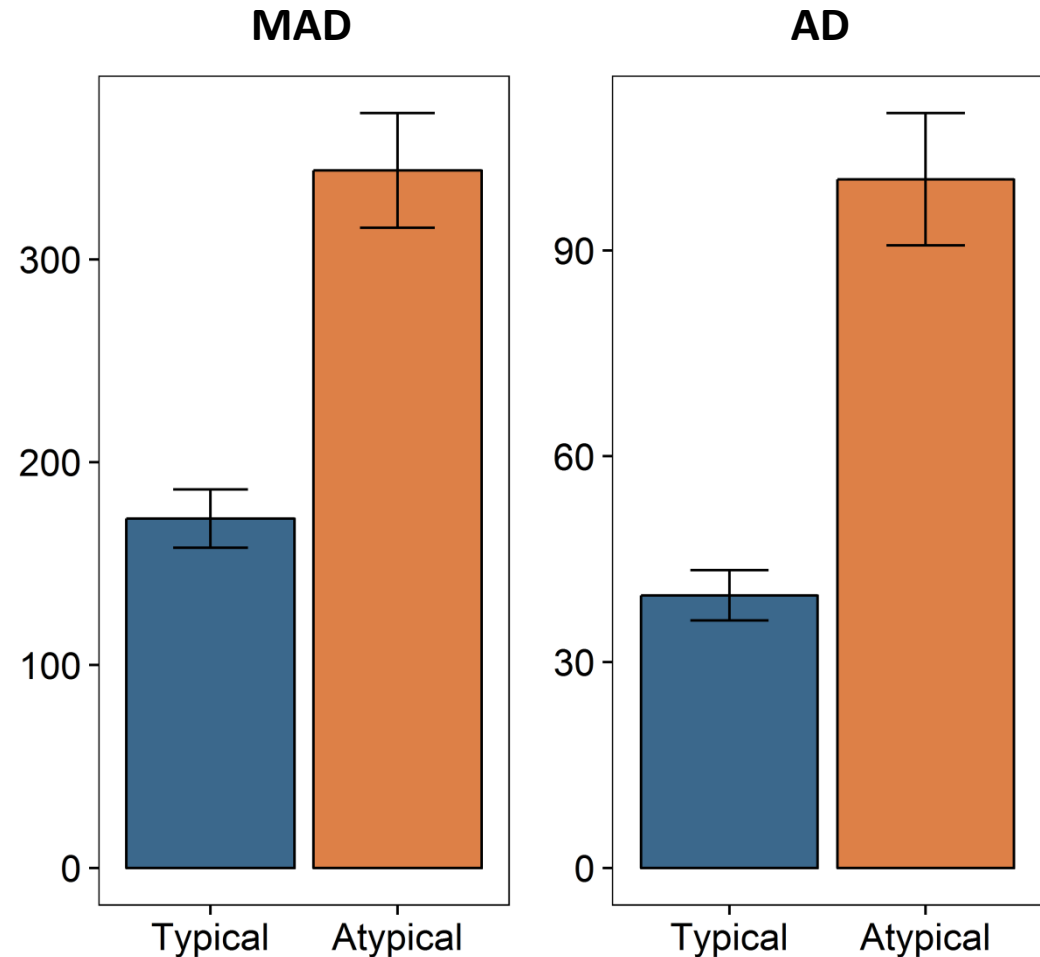
22

### □ MAD larger for atypical exemplars

- $d_z = 0.87, p < .001$
- $BF_{10} = 1.57 * 10^6$

### □ AD larger for atypical exemplars

- $d_z = 0.87, p < .001$
- $BF_{10} = 1.78 * 10^6$



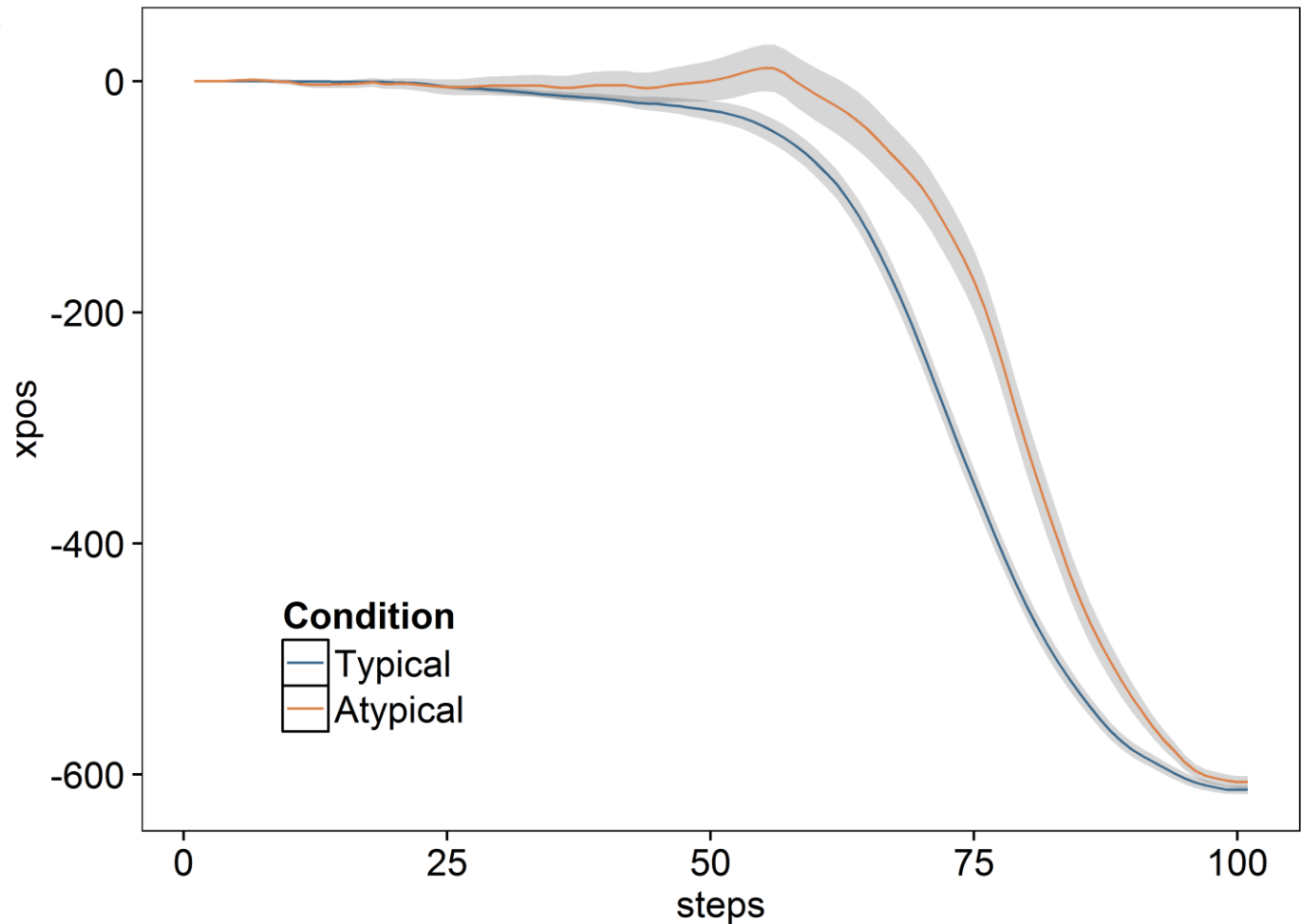
Error bars represent 1 SEM.

# Implementation & analysis

## Average x-positions per time step

23

- Similar analysis by Dale et al.: Paired  $t$ -test of average time-normalized x-position per per time step
- Significant differences ( $p < .05$ ) between conditions from time steps 54 to 95



Error bars represent 1 SEM.



# Implementation & analysis

## Selected mouse-tracking measures

24

Measure	Definition	Possible interpretation	Example
Maximum absolute deviation (MAD)	Maximum deviation from idealized trajectory	Maximum attraction of non-chosen option	McKinstry et al. (2008)
Average Deviation (AD)	Mean deviation from idealized trajectory	Average attraction of non-chosen option	Koop & Johnson (2011)
Area under curve (AUC)	Geometric area between actual and idealized trajectory	Total attraction of non-chosen option	Spivey et al. (2005)
x-flips (xpos_flips)	Number of directional changes along x-axis	Instability, reversal of the momentary valence	Koop & Johnson (2013)
x-reversals (xpos_reversals)	Number of crossings of the y-axis	General reversal of preference	Koop & Johnson (2013)

# Implementation & analysis

## Analytical and theoretical challenges

25

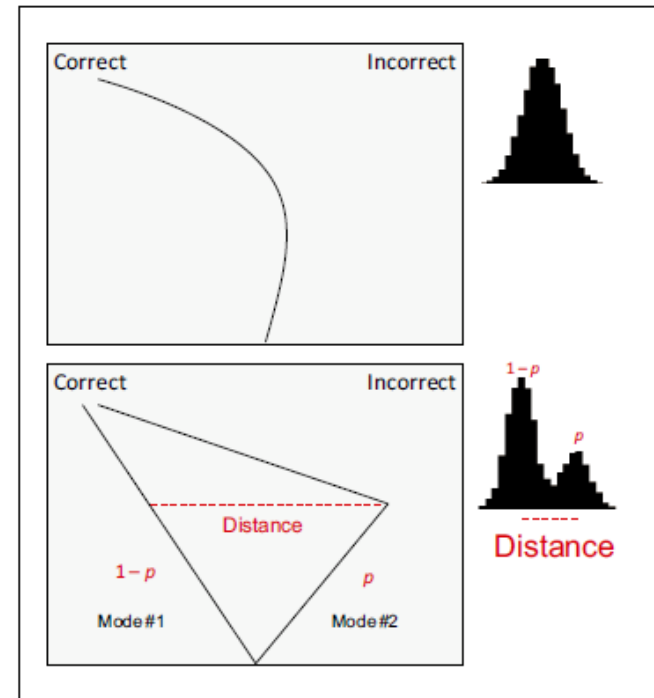
- Interpretation of measures still needs to be validated
- Multitude of mouse-tracking measures available
  - ▣ Often highly correlated in practice
  - ▣ There is no standard yet which measure should be used
    - ensure that result does not depend on the specific measure used
    - decide which is the measure of interest a priori / conduct pre-registered replications of your findings
- Consider effects of aggregation by inspecting distribution of trajectories and indices on the trial level

# Implementation & analysis

## Smooth competition vs. abrupt shifts

26

- Different assumptions about response process (e.g., Hehman et al., 2015)
  - ▣ Single process
    - “smooth graded competition” in all trials
    - Continuous competition between response options
  - ▣ Dual process
    - “abrupt shifts” / **Change of Mind** in some trials: Initial movement towards one option, then reversal and choice of other option
    - Straight movements in other trials
- Statistical analysis of AUC or MAD distribution (Freeman & Dale, 2013)
  - ▣ “smooth graded competition” → unimodal
  - ▣ “abrupt shifts” → bimodal



# Implementation & analysis

## Methods for assessing bimodality and trajectory shapes

27

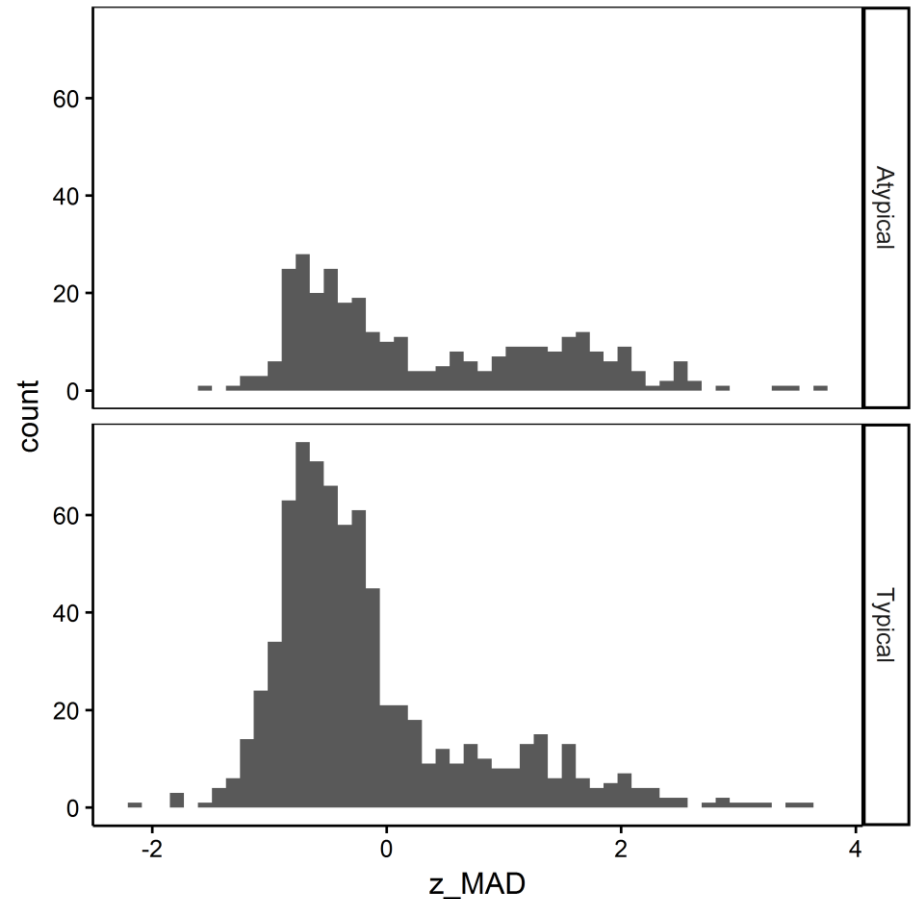
- Bimodality coefficient (**BC**, e.g., Pfister et al., 2013)
  - ▣ 
$$BC = \frac{m_3^2 + 1}{m_4 + 3 \cdot \frac{(n-1)^2}{(n-2)(n-3)}}$$
  - ▣ Bimodal, if **BC > 0.555**
- Hartigan's dip statistic (**HDS**, Hartigan & Hartigan, 1985)
  - ▣ Statistical test (H0: Distribution is unimodal)
  - ▣ If  **$p < .05$** , distribution is multimodal (i.e., at least bimodal)

# Implementation & analysis

## Assessment of bimodality

28

- Distribution of standardized MAD
- **Bimodality coefficient ( $BC$ )**
  - ▣  $BC_{\text{typical}} = .61$ ;  $BC_{\text{atypical}} = .59$
  - ▣ Indicates bimodality as  $BC > .555$
- Also influenced by setup of study (cf. design factors)



# Implementation & analysis

## Assessing distribution of individual trajectory shapes

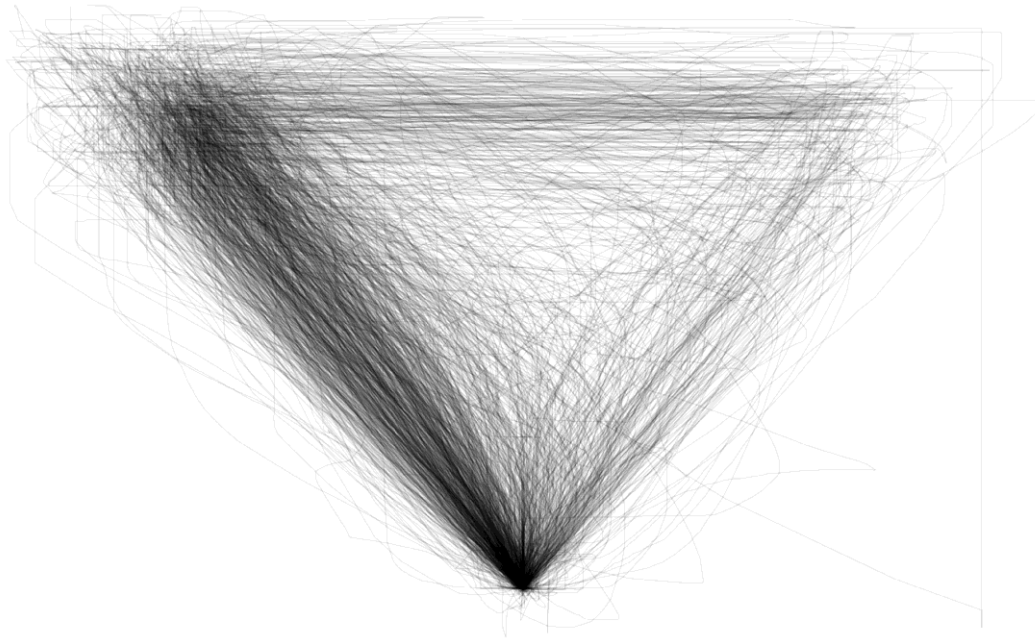
29

- Assess distribution of trajectory shapes (Wulff et al., in press)
  - ▣ **Bimodality analyses** so far focus on a **single measure** only
  - ▣ New analyses proposed taking complete **trajectory shape** into account
  - ▣ General question: is aggregate trajectory representative of individual trajectories – or are there **different types** of trajectories on the trial level?
- Visualization tools
  - ▣ Animations
  - ▣ Heatmaps and difference maps
- Analyses tools
  - ▣ Clustering
  - ▣ Prototype allocation

# Implementation & analysis

## Heatmap of raw trajectories

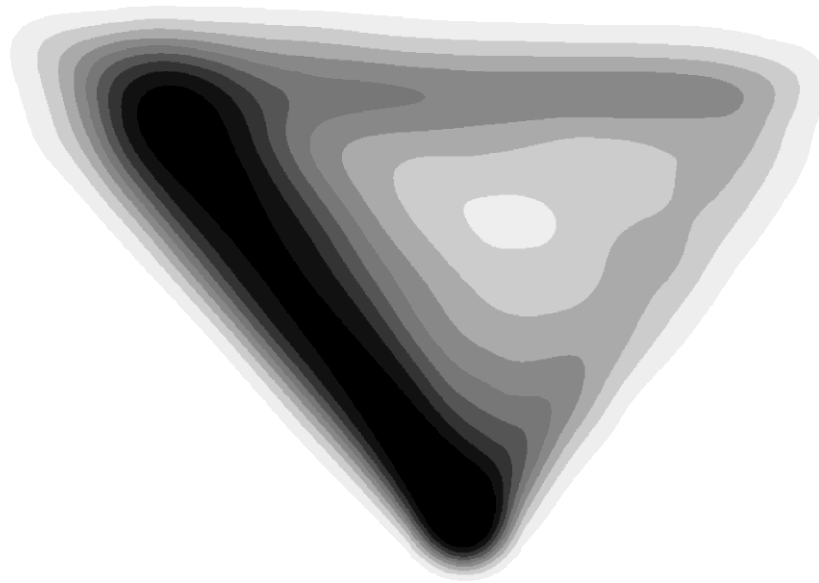
30



# Implementation & analysis

Heatmap of raw trajectories (smoothed)

31

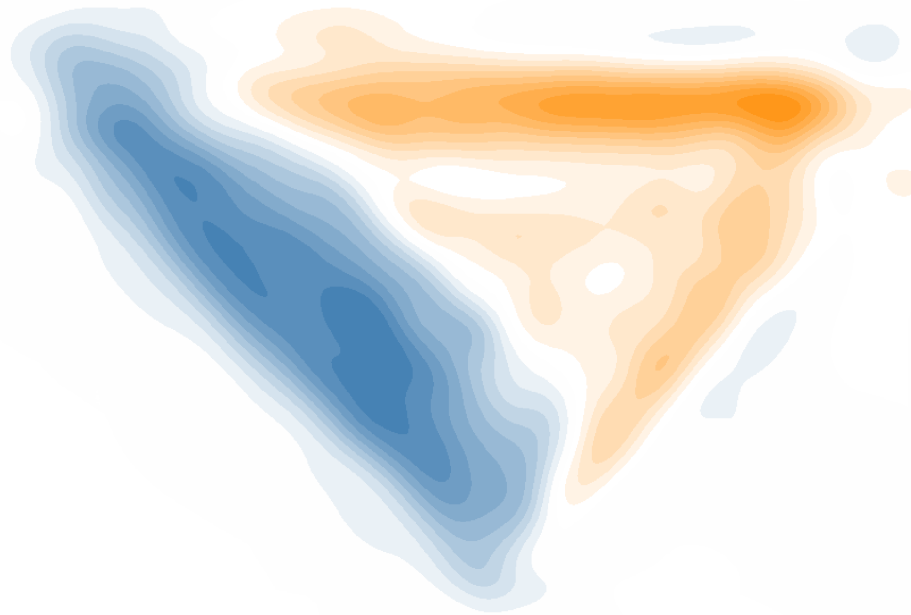




# Implementation & analysis

Difference map for typical vs. atypical condition

32



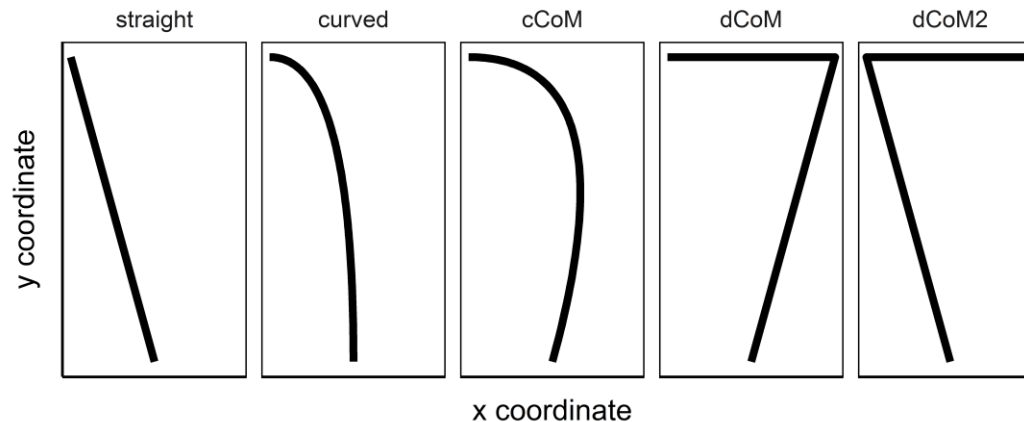
# Implementation & analysis

## Prototype recognition (Wulff et al., in press)

33

### □ Specify set of prototypes

- ▣ Set of prototypes based on clustering results of the meta-analysis by Wulff et al. (in prep.)



### □ Spatialize trajectories

- ▣ Resample trajectories to small number of points distributed equally across space

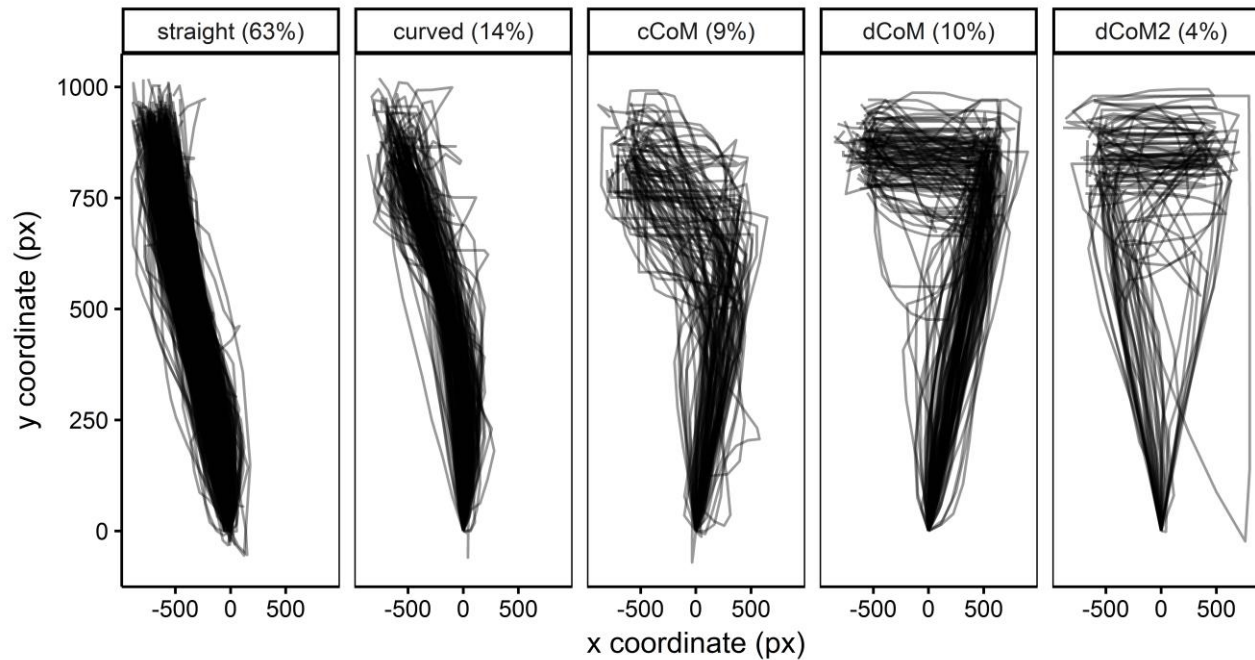
### □ Assign trajectories to prototypes

- ▣ Compute dissimilarity between every trajectory and prototype
- ▣ Assign trajectory to prototype with smallest distance
- ▣ (Potentially exclude trajectories where smallest distance is too large)

# Implementation & analysis

## Prototype allocation for replication experiment

34

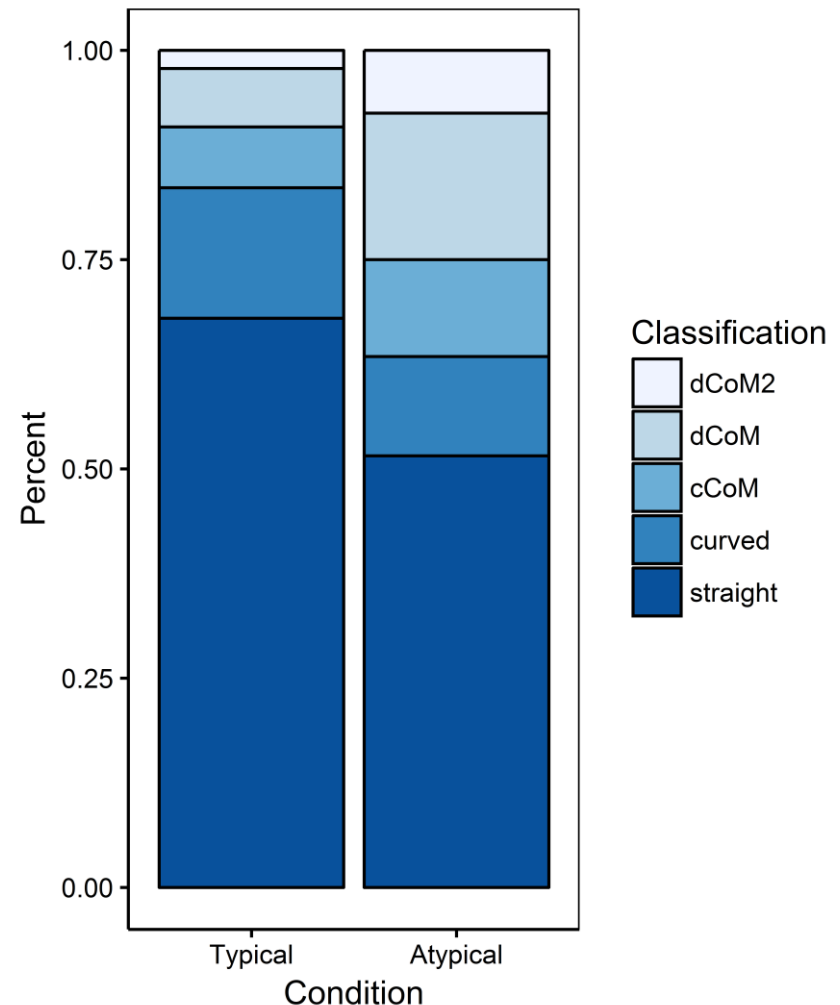


# Implementation & analysis

## Classification frequencies per condition

35

- Relative frequency of prototype classification differs for conditions
  - ▣  $\chi^2 = 57.9, p < .001$
- Atypical condition predicts occurrence of types that indicate more conflict
  - ▣ in ordinal mixed regression model on trial level
  - ▣ with random intercept per participant
  - ▣  $z = 6.74, p < .001$





# Previous applications

Focusing on JDM research

# Mouse-tracking

## Application domains

37

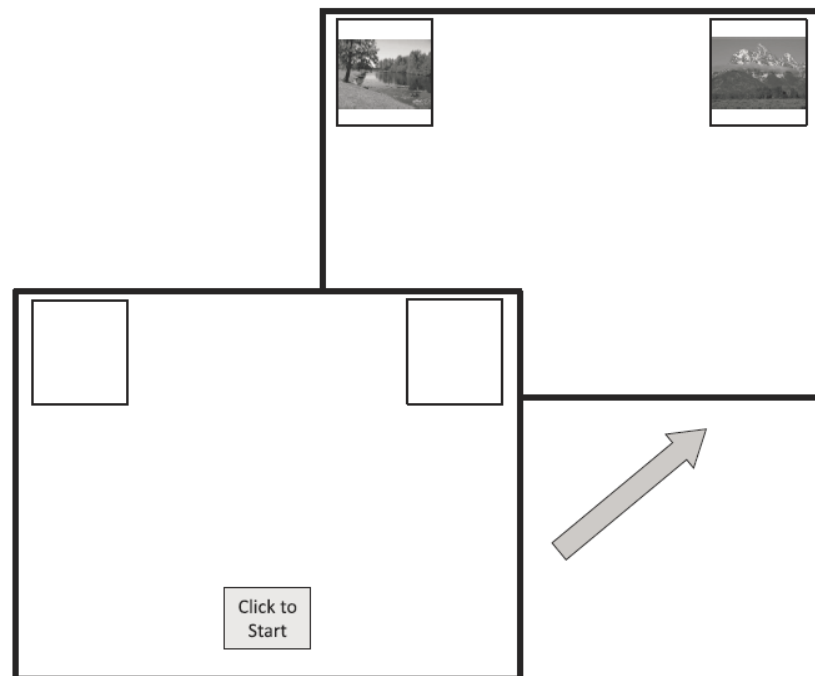
- Application of mouse-tracking in a **growing number of psychological domains** (Reviews by Freeman, in press; Stillman et al., 2018)
  - ▣ Semantic processing (e.g., Spivey et al., 2005; Dale & Duran, 2011)
  - ▣ Social cognition (e.g., Freeman et al., 2008; Freeman & Ambady, 2011)
  - ▣ Learning and memory (e.g., Dale et al., 2008; Koop & Criss, 2016)
  - ▣ Self-control (e.g., Sullivan et al., 2015; Stillman et al., 2017)
  
- In the last years also extended to **JDM research**
  - ▣ Intertemporal choice (Dshemuchadse et al., 2013)
  - ▣ Moral dilemmas (Koop, 2013)
  - ▣ Decisions under risk (Koop & Johnson, 2013)
  - ▣ Social dilemmas (Kieslich & Hilbig, 2014)
  - ▣ Judgmental biases (Szasz et al., 2018; Travers et al., 2016)

# Preferential decision making

## Validation experiment (Koop & Johnson, 2013, Exp. 1)

38

- Decisions between affective images
  - ▣ Task: Which of two images do you prefer?
  - ▣ Pictures from IAPS database: provides norms for pleasantness ratings
  - ▣ Creation of pairs where difference in preferences is systematically varied

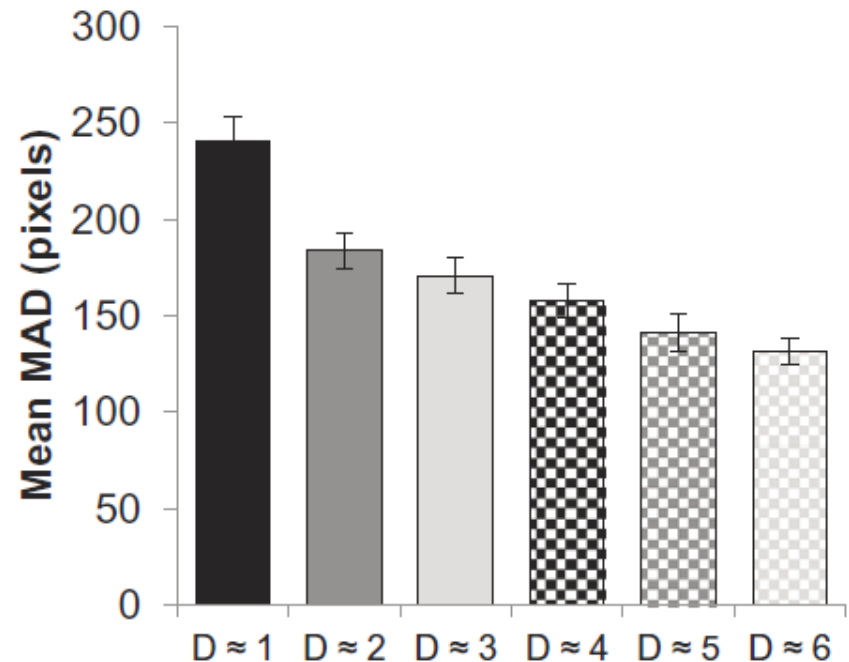
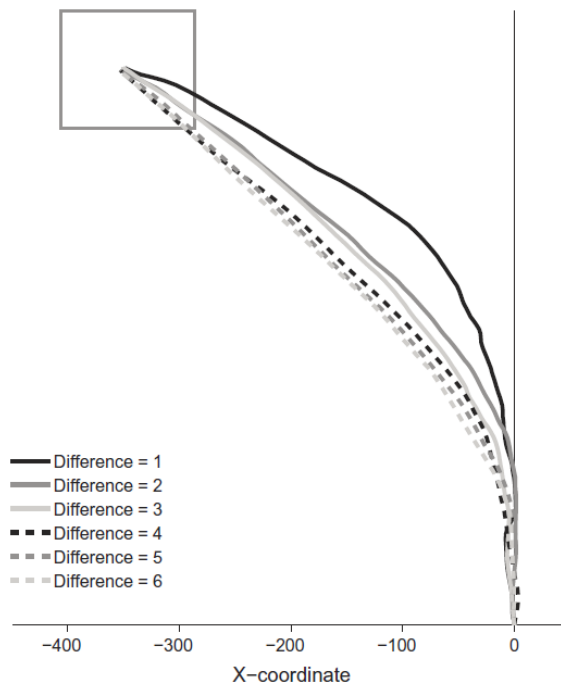


# Preferential decision making

## Validation experiment (Koop & Johnson, 2013, Exp. 1)

39

- Decisions between affective images
  - ▣ Increase in difference of a priori preference ratings leads to
    - Decrease in trajectory curvature
    - Decrease in maximum absolute deviation (MAD)





# Social dilemmas

## Basic structure

40

- Social dilemma (Dawes 1980; Van Lange et al., 2013)
  - ▣ Individuals can choose between two options
    - Defection
    - Cooperation
- Standard social dilemma: **Prisoner's dilemma game**  
(PDG; Rapoport & Chammah, 1965)

		Player 1	
		cooperates	defects
Player 2	cooperates	100   100	200   0
	defects	0   200	50   50

# Social dilemmas

## Spontaneous cooperation?

41

- **Theoretical proposition** (Rand et al., 2012, 2014)
  - ▣ People are **spontaneously** inclined to **cooperate**
  - ▣ **Defection** requires effortful **deliberation**
- **Empirical test using response times**
  - ▣ Idea: **spontaneous = fast**, **deliberative = slow**
  - ▣ Mixed results (e.g., Rand et al., 2014; meta-analysis by Rand, 2016; Registered replication report, 2017)
  - ▣ Other factors may influence speed (e.g., guessing, information search)
- **Experiment using mouse-tracking** (Kieslich & Hilbig, 2014)
  - ▣ When deciding to **defect**, mouse movements should be **more curved** towards non-chosen option (cooperation)
  - ▣ When deciding to **cooperate**, mouse movements should be **less curved** towards non-chosen option (defection)

# Social dilemmas

## Mouse-tracking experiment (Kieslich & Hilbig, 2014)

42

- Lab experiment ( $N = 115$ )
  - ▣ at the University of Mannheim
  - ▣ implementation in OpenSesame (Mathôt et al., 2012) in combination with
    - mousetrap plug-ins for mouse-tracking (Kieslich & Henninger, 2017)
    - psynteract plug-ins for interactive experiments (Henninger, Kieslich, & Hilbig, 2017)
- Participants play 15 two-person social dilemma games
  - ▣ without receiving feedback
  - ▣ random order
  - ▣ incentivized (5 interactions paid out,  $\emptyset$  payout: 2.56 €)
- Social dilemma games
  - ▣ 5 x prisoner's dilemma game (PDG)
  - ▣ 5 x chicken game
  - ▣ 5 x stag hunt game

Option A

Option B

Decision 9 of 15

You choose Option A

You choose Option B

Person 2 chooses Option A

100 | 100

200 | 0

Person 2 chooses Option A

Person 2 chooses Option B

0 | 200

50 | 50

Person 2 chooses Option B

You choose Option A

You choose Option B

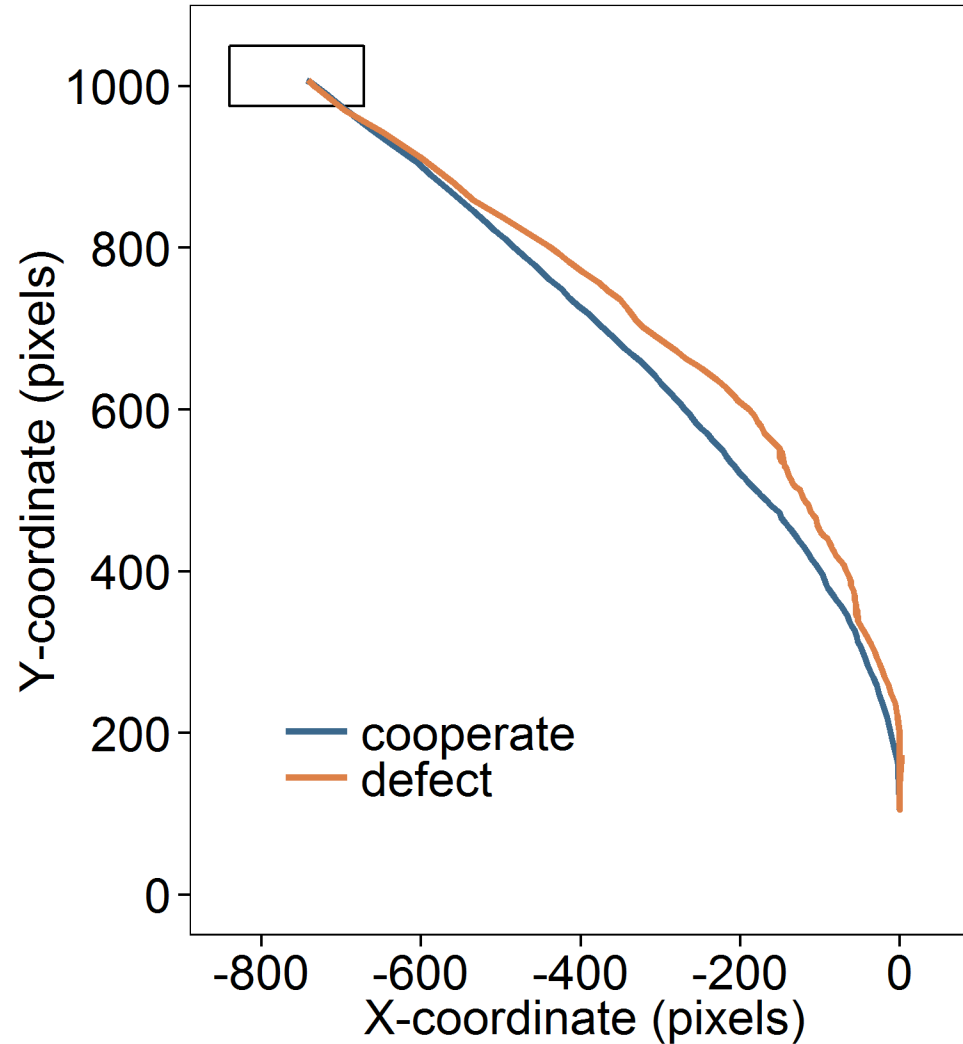
Please choose  
between Option A and B.

Start

# Social dilemmas

## Average time-normalized response trajectories

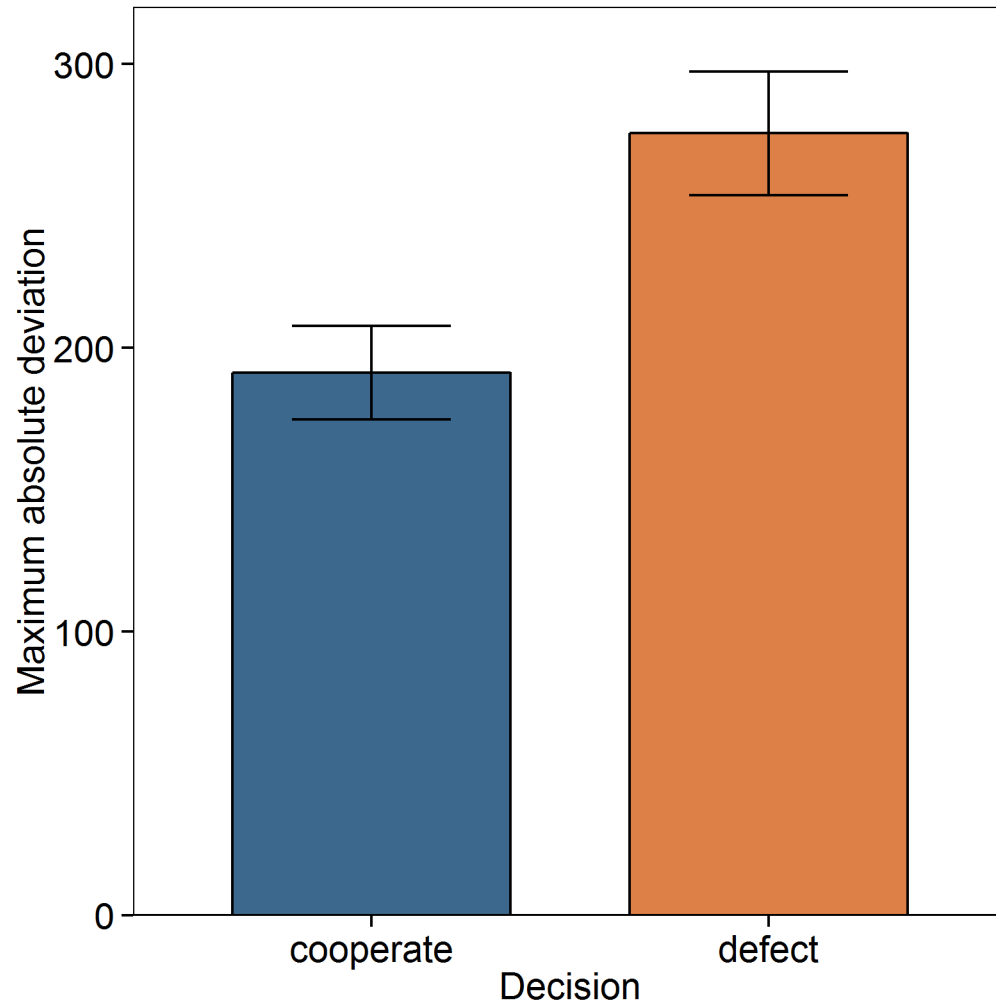
44



# Social dilemmas

## Maximum deviation per decision

45



### □ Main effect of decision

- ▣ MAD significantly higher for defection than for cooperation

### □ Effect replicated

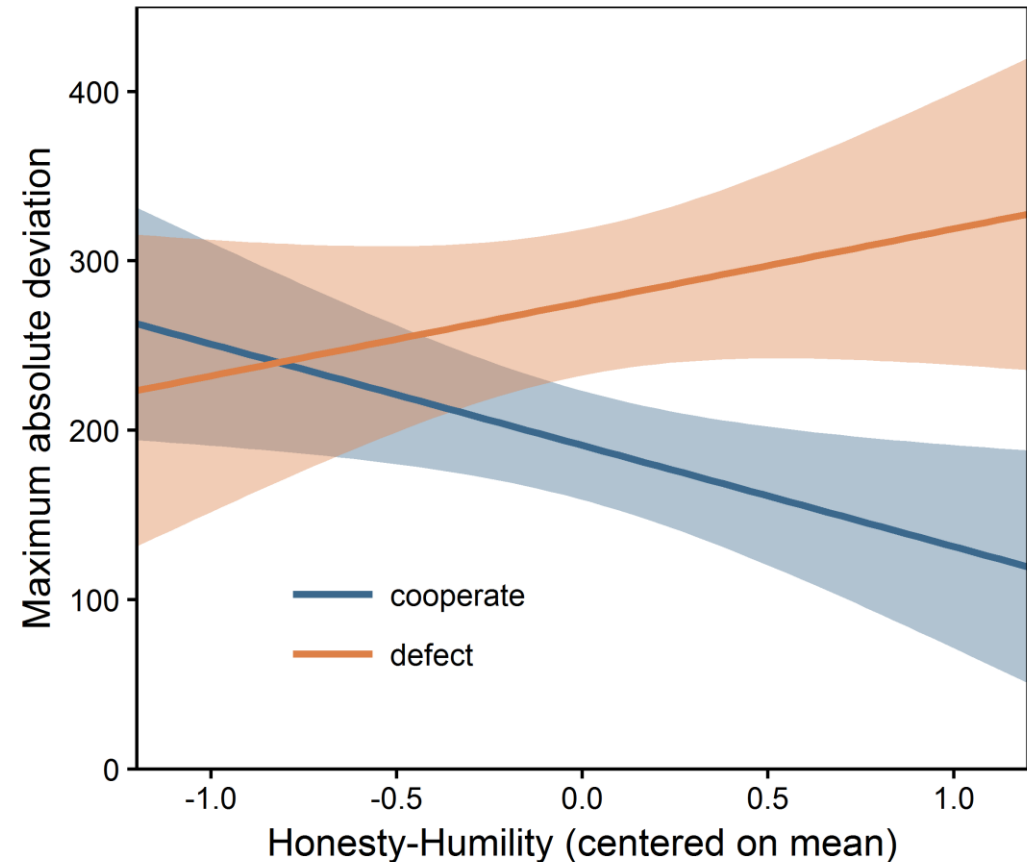
- ▣ With different measures
- ▣ With filtered trials
- ▣ With linear mixed model on trial level

# Social dilemmas

## Predicting individual differences in conflict

46

- Individual differences in conflict: Differences should be stronger for individuals high in Honesty-Humility
  - ▣ Dispositional cooperativeness
  - ▣ Basic personality factor in the HEXACO personality model (Ashton & Lee, 2007)
- Significant interaction between HH and decision



Confidence bands represent 95% CI.

# Social dilemmas

## Mouse-tracking challenges

47

### □ Experimental control over comparison dimension

- Mouse-tracking tasks usually involves “correct”/desired response option + comparison dimension is experimentally manipulated
- Here **final choice** constitutes **comparison dimension of interest**
  - loss of experimental control
  - use of different games to achieve variation in cooperation rates

### □ Complexity and amount of information

- **Amount of information** and **complexity of decision** considerably higher than in previous tasks
- Mouse movements more noisy (e.g., reading movements in some trials)
  - Current solution: analyses replicated with and without problematic trials
  - Ideal solution: simpler task design with less information
    - working on conceptual replication in binary public goods game, also taking into account the newly proposed analytical approaches (prototype mapping)



# Action selection

## Simon effect and conflict adaptation (Scherbaum et al., 2010)

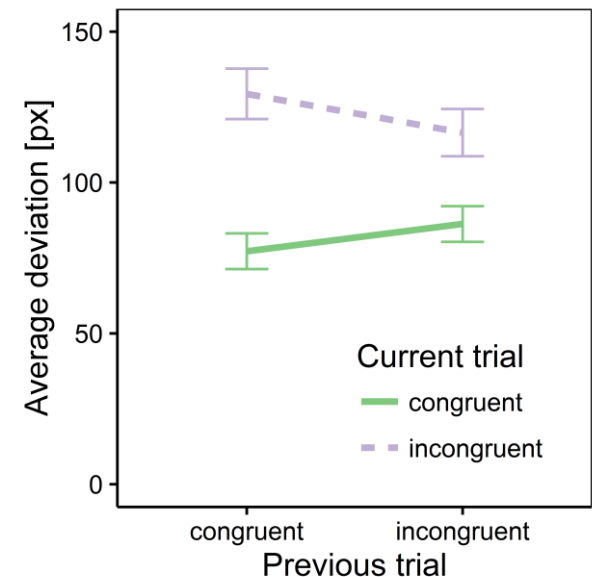
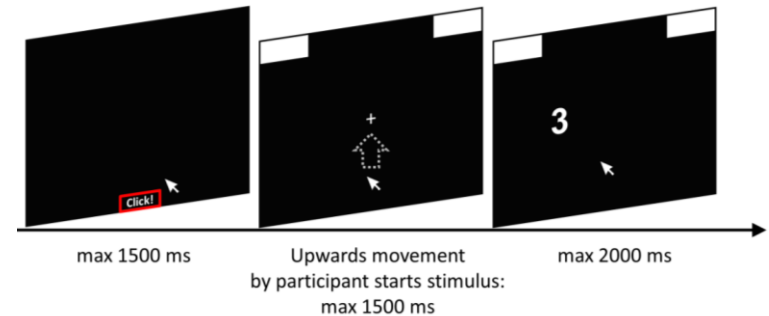
48

### □ Mouse-tracking in Simon task

- Participants have to **click on left vs. right** option depending on the stimulus (e.g., left if number < 5, otherwise right)
- Position of stimulus varied (left vs. right) so that desired response and position are either **congruent** or **incongruent**

### □ Results

- **Simon effect**: larger deviations in incongruent than in congruent trials
- **Conflict adaptation**: Simon effect reduced if previous trial was incongruent



# Action selection

## Time continuous multiple regression (Scherbaum et al., 2010)

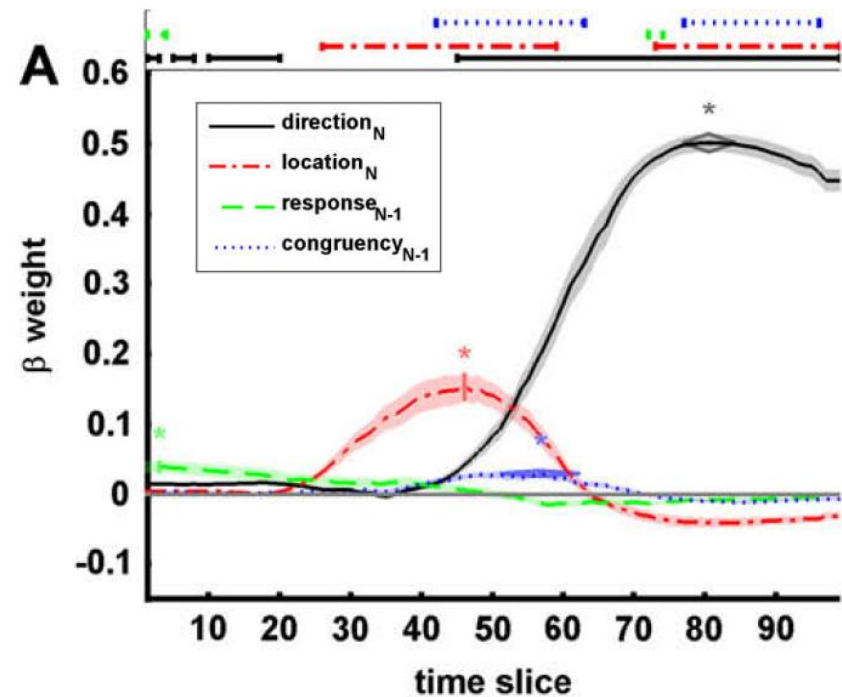
49

### Time continuous multiple regression

- Criterion: mouse movement angles on the XY plane ( $\approx$  **movement direction**)
- Separate regressions per time step and participant
- Reveals **temporal order** and strength with which each predictor influences preference development

### Predictors

- Task relevant
  - **Direction** (left/right)
- Task irrelevant
  - **stimulus location** (left/right)
  - **previous response** (left/right)
  - **congruency sequence** (same/different)



Average  $\beta$  weights per time step and predictor.

# Design factors

## Overview

50

- Researchers face a number of **design choices** when creating mouse-tracking experiments
  - ▣ Starting procedure (static, restricted initiation time, dynamic)
  - ▣ Cursor speed settings (velocity & acceleration)
  - ▣ Indicate response via click vs. touch
- Some authors have given **recommendations** about designing mouse-tracking studies (Fischer & Hartmann, 2014; Hehman et al., 2015)
- Empirical **validation** studies are being conducted (Scherbaum & Kieslich, in press; Kieslich et al., in preparation)

# Design factors

## Preliminary summary of findings

51

### □ Response indication

- **Click** on button leads to larger effects than **touch** – effect related to higher proportion of trials with extreme movements to non-chosen option

### □ Mouse sensitivity settings

- Did not significantly influence effect of interest in static setup – although **default** settings generally lead to more extreme curvature than **reduced mouse speed**
- Reducing mouse speed becomes relevant for dynamic start condition to ensure stimulus information can be acquired during upwards movement

### □ Starting procedure

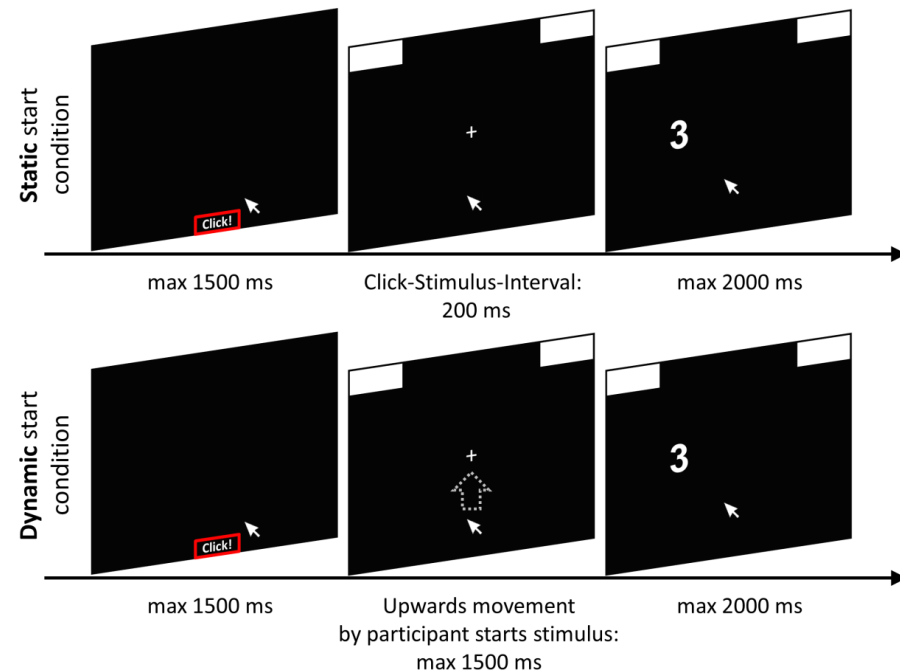
- Restricting **maximum initiation time** led to larger effects – a **dynamic start** only influenced shape but not effect size
- However, restricting initiation times also led to largest proportion of excluded trials (and seemed to be challenging for some participants)

# Starting procedure: Static vs. dynamic start

## Method (Scherbaum & Kieslich, in press)

52

- Mouse-tracking in Simon task
  - ▣ Participants **click on left vs. right** option depending on stimulus (left if number < 5, otherwise right)
  - ▣ Position of stimulus varied (left vs. right) so that desired response and position are either **congruent** or **incongruent**
- Variation starting procedure
  - ▣ **Dynamic**: move upwards to display stimulus (data from Scherbaum et al., 2010)
  - ▣ **Static**: stimulus displayed after fixed interval of 200 ms (typical duration of movement initiation in dynamic condition) (new data)

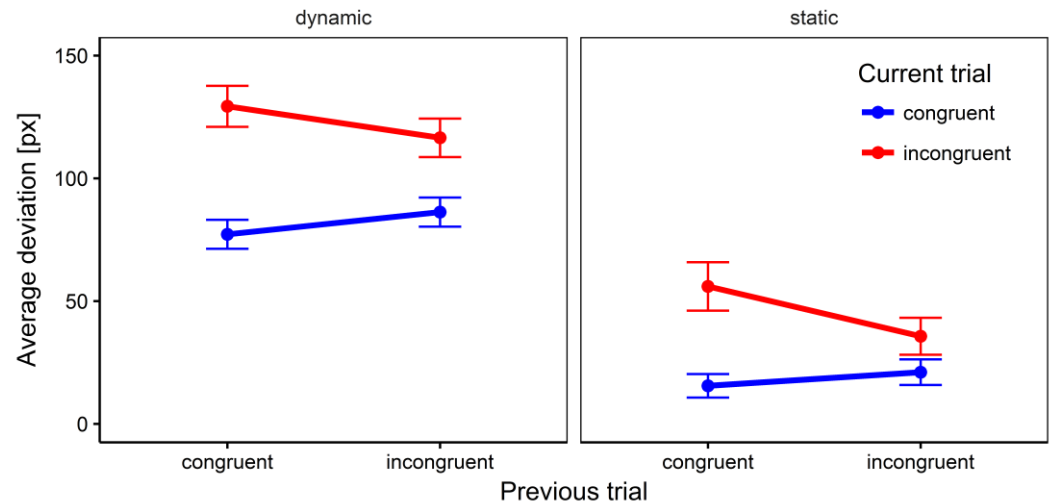


# Starting procedure: Static vs. dynamic start

## Discrete effects: Results for average deviation

53

- Simon effect and congruency sequence effect replicated in both conditions
- No significant interaction of theoretically important effects with starting procedure



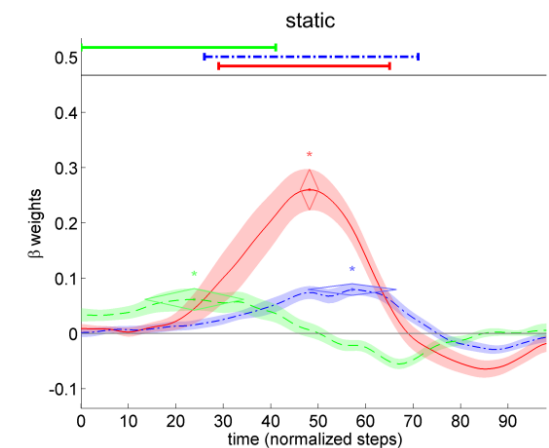
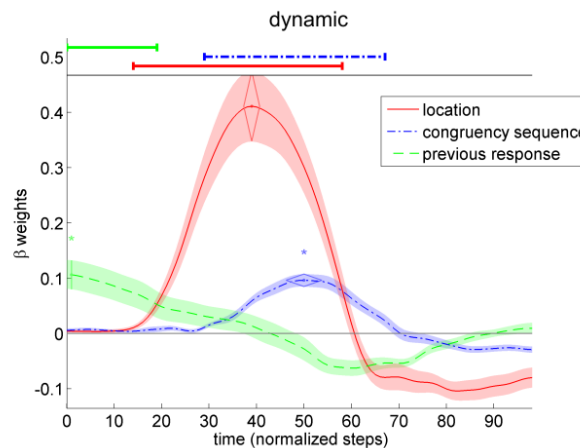
Error bars represent 1 *SEM*.

# Starting procedure: Static vs. dynamic start

## Dynamic effects: Time-continuous angle regression

54

- Time continuous multiple regression predicting vertical movement angle at each time point
- Predictors
  - ▣ location (congruency)
  - ▣ congruency sequence (same / different)
  - ▣ previous response (same / different)
- Effects stronger and more temporarily distinct in dynamic starting condition



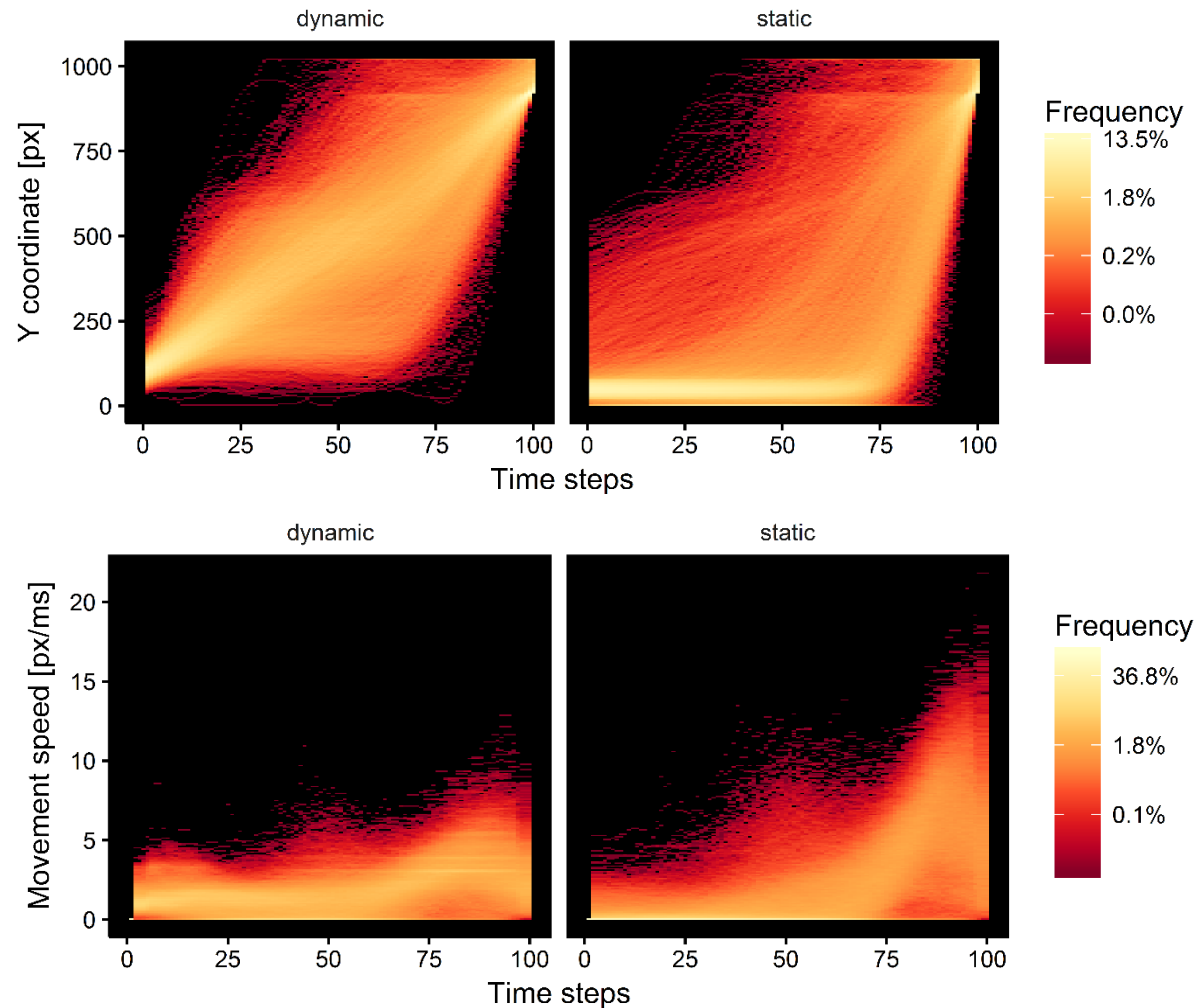
Average  $\beta$  weights per time step and predictor.  
Lines indicate segments of  $\beta$  weights significantly  $> 0$ .

# Starting procedure: Static vs. dynamic start

## Movement consistency

55

- Smooth and consistent upwards movement in dynamic starting condition
- Participants in static starting condition often stay at bottom of screen for more than half of the trial before moving upwards quickly

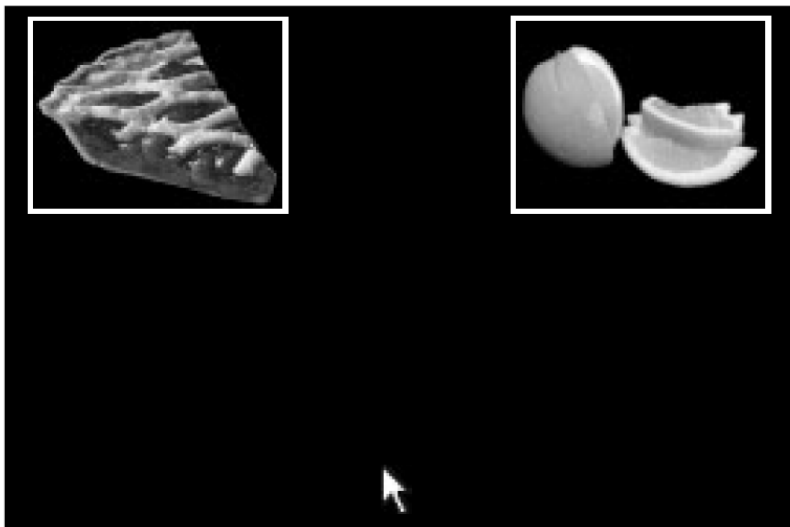




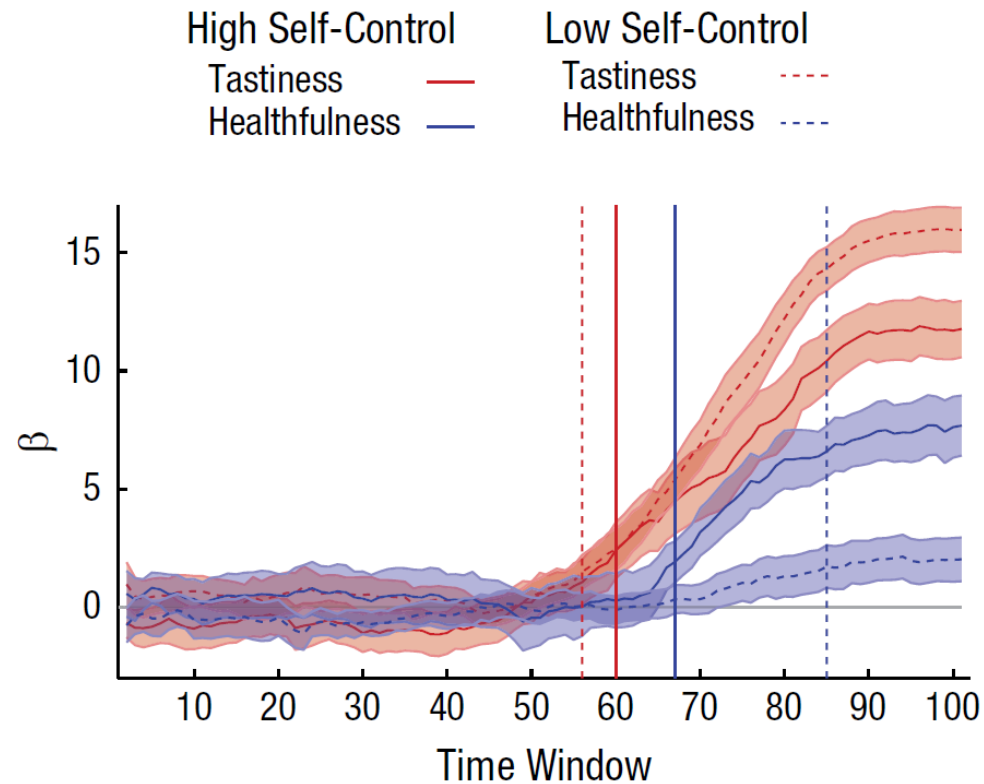
# Self-control

## Food choices (Sullivan et al., 2015)

56



Choice Screen



# Decisions under risk

## Basic paradigm

57

- Risky choice / decisions under risk
  - ▣ Which of the two gambles do you want to play?

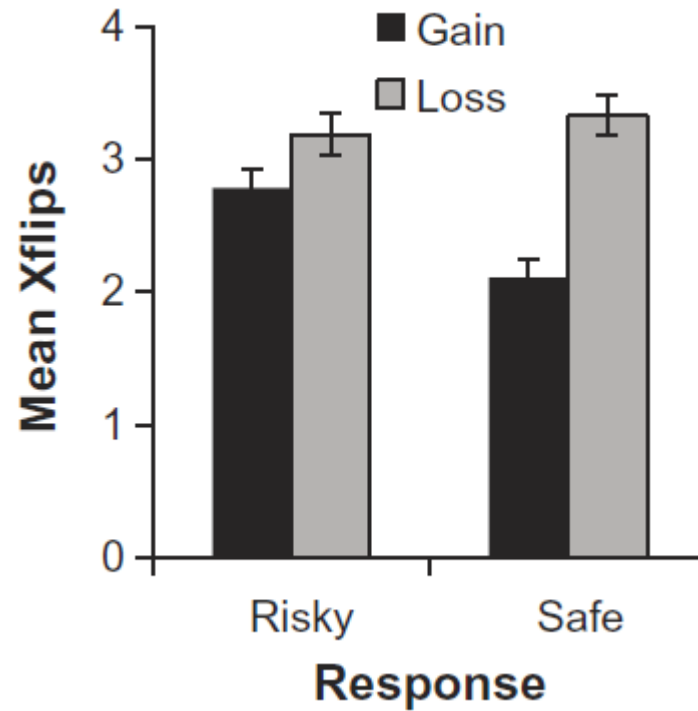
Gamble A	Gamble B
You have a 50% chance of winning \$90, otherwise nothing	You have a 90% chance of winning \$50, otherwise nothing

- Gamble A: “risky”
  - ▣ Higher amount, lower probability of winning
- Gamble B: “safe”
  - ▣ Lower amount, higher probability of winning

# Decisions under risk

x-flips (Koop & Johnson, 2013, Exp. 2)

58



# Decisions under risk

## Combining mouse- and eye-tracking (Koop & Johnson, 2013, Exp. 3)

59

- Change in x-position ( $\Delta x$ ) as function of **transitions of attention**

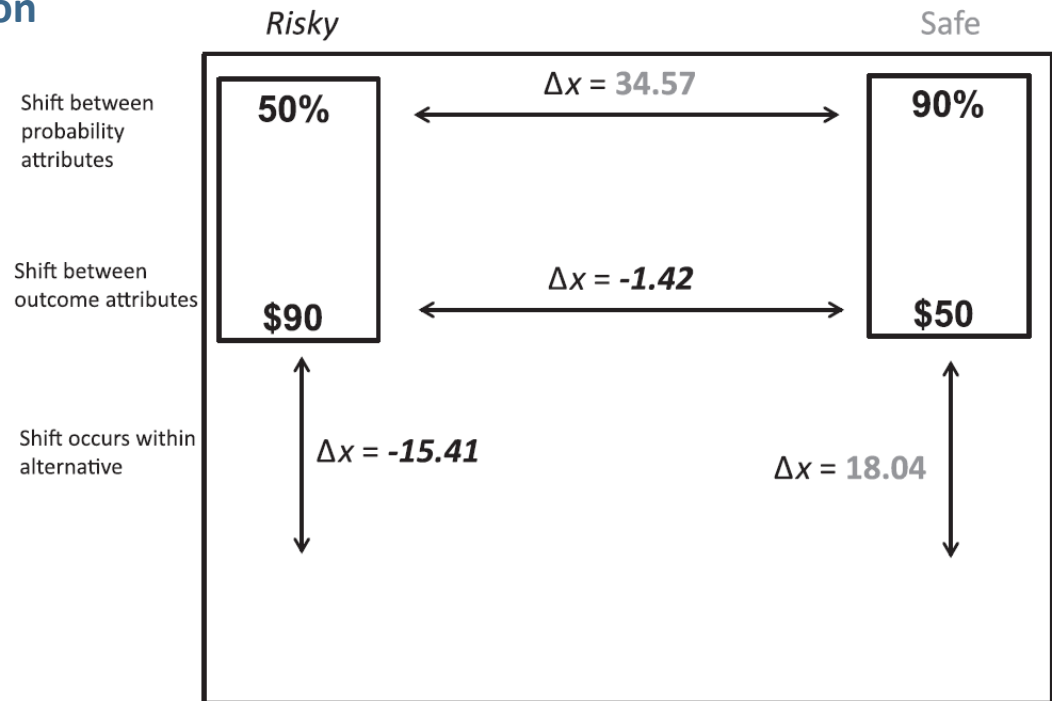
- $\Delta x > 0$ : movement towards safe gamble
- $\Delta x < 0$ : movement towards risky gamble

- **Evidence accumulation model**

- Predict **momentary preference** based on **visual input**
- Mean correlation between predicted preference and x-position is  $r = .78$

- **Conclusions**

- **Visual attention** to probability and outcome information **predicts mouse response**
- Mouse movements largely reflect quality of acquired information





Your experiments

# Mouse-tracking introduction (Monday)

61

- 13:00-14:30 General introduction to mouse-tracking
  - ▣ Paradigm and assumptions
  - ▣ Implementation and analysis
  - ▣ Previous applications
- 14:30-15:00 Introduction to task
  - ▣ Type of experiments considered
  - ▣ Your tasks during the workshop
- 15:00-17:00 Develop experimental design conceptually
- 17:00-18:00 Present experimental design in plenum

# Your tasks during the workshop

- Goal of workshop
  - ▣ Design, build, pre-register, run, and analyze a mouse-tracking experiment
  - ▣ In small groups
- Monday
  - ▣ Develop experimental design (task, manipulation, hypotheses, measures)
  - ▣ Present experimental design in plenum
- Tuesday
  - ▣ Build experiment
  - ▣ Register experiment at OSF
  - ▣ Participate in experiments
- Wednesday
  - ▣ Analyze and visualize your data
  - ▣ Discuss your results
- Saturday
  - ▣ Present results

# Type of experiments

63

- In the experiment, participants complete a number of trials that involve decisions of the same structure
- In each trial, participants have to decide between two options by clicking on the corresponding button (two-alternative forced choice task, 2AFC)
- Between trials, the stimulus to be decided upon varies (usually) and / or the two response categories
- The stimulus (and/or the response options - in case they vary) should be simple (e.g., a single word, a picture)



# Implementation & analysis

## Software

64

- **Custom extensions** for experimental software
  - ▣ Code based implementations, e.g., in E-Prime or MATLAB
  - ▣ Also need scripts for preprocessing the data
  - ▣ Require **programming** skills
- **MouseTracker** (Freeman & Ambady, 2010)
  - ▣ **Stand-alone** program
  - ▣ Relatively easy to use, but limited in features and flexibility
  - ▣ Free of charge but closed source, Windows only



- **Mousetrap** (Kieslich & Henninger, 2017; Kieslich, Wulff et al., in preparation)
  - ▣ Drag & drop plugins for experimental software **OpenSesame**
  - ▣ **R package mousetrap** for preprocessing and analysis
  - ▣ Open source, free of charge, cross-platform
  - ▣ Available from <http://pascalkieslich.github.io/mousetrap/>



# Software for the workshop

65

- To create mouse-tracking experiments, first install OpenSesame. It is available from <http://osdoc.cogsci.nl/3.2/download/>.
- To install the mousetrap plugin for OpenSesame, follow the instructions at <https://github.com/pascalkieslich/mousetrap-os#installation>. Please make sure to install the latest version of OpenSesame (3.2.4) and the development version of the mousetrap-os plugin.
- To analyze mouse-tracking data install R (<https://www.r-project.org/>) and RStudio (<https://www.rstudio.com/products/rstudio/download/>).
- Afterwards, please run the following command in R to install the required packages:  
`install.packages(c("readbulk", "mousetrap"))`
- Screen resolution of experiment for lab computers: 1280 x 1024 px

# Thank you!

Questions and comments are highly appreciated!

Now & via email: [kieslich@psychologie.uni-mannheim.de](mailto:kieslich@psychologie.uni-mannheim.de)  
[dirk.wulff@gmail.com](mailto:dirk.wulff@gmail.com)

Mousetrap-os plugins: <https://github.com/pascalkieslich/mousetrap-os>

Mousetrap R package: <http://pascalkieslich.github.io/mousetrap/>

Thanks:

Felix Henninger, co-developer of mousetrap-os plugin and R package

Jonas Haslbeck & Michael Schulte-Mecklenbeck, co-developers of mousetrap R package

Mila Rüdiger and Monika Wiegelmann for data collection and testing



## **DAY 2:**

# **CREATING MOUSE-TRACKING EXPERIMENTS**

Pascal Kieslich (University of Mannheim)

Workshop at the EADM Summer School 2018 in Salzburg, Austria

# Creating mouse-tracking experiments (Tuesday)

68

- 09:00-11:00 OpenSesame & mousetrap-os introduction
- 11:00-12:00 Build experiment
- 12:00-13:00 Lunch break
- 13:00-15:00 Build experiment
- 15:00-16:00 Register experiment at OSF (Michael)
- 16:00-17:00 Keynote Neil Stewart
- 17:00-18:00 Meet the Scientist
- 18:00-19:00 Participate in experiments (Lab computers)

# Software for the workshop

69

- To create mouse-tracking experiments, first install OpenSesame. It is available from <http://osdoc.cogsci.nl/3.2/download/>.
- To install the mousetrap plugin for OpenSesame, follow the instructions at <https://github.com/pascalkieslich/mousetrap-os#installation>. Please make sure to install the latest version of OpenSesame (3.2.4) and the development version of the mousetrap-os plugin.
- To analyze mouse-tracking data install R (<https://www.r-project.org/>) and RStudio (<https://www.rstudio.com/products/rstudio/download/>).
- Afterwards, please run the following command in R to install the required packages:  
`install.packages(c("readbulk", "mousetrap"))`
- Screen resolution of experiment for lab computers: 1280 x 1024 px



# OpenSesame & mousetrap-os introduction

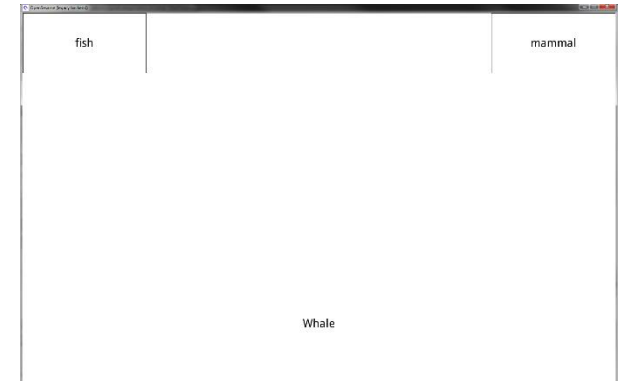
# Creating mouse-tracking experiments

## Replication study of Dale, Kehoe, & Spivey (2007)

71

### □ Animal categorization task

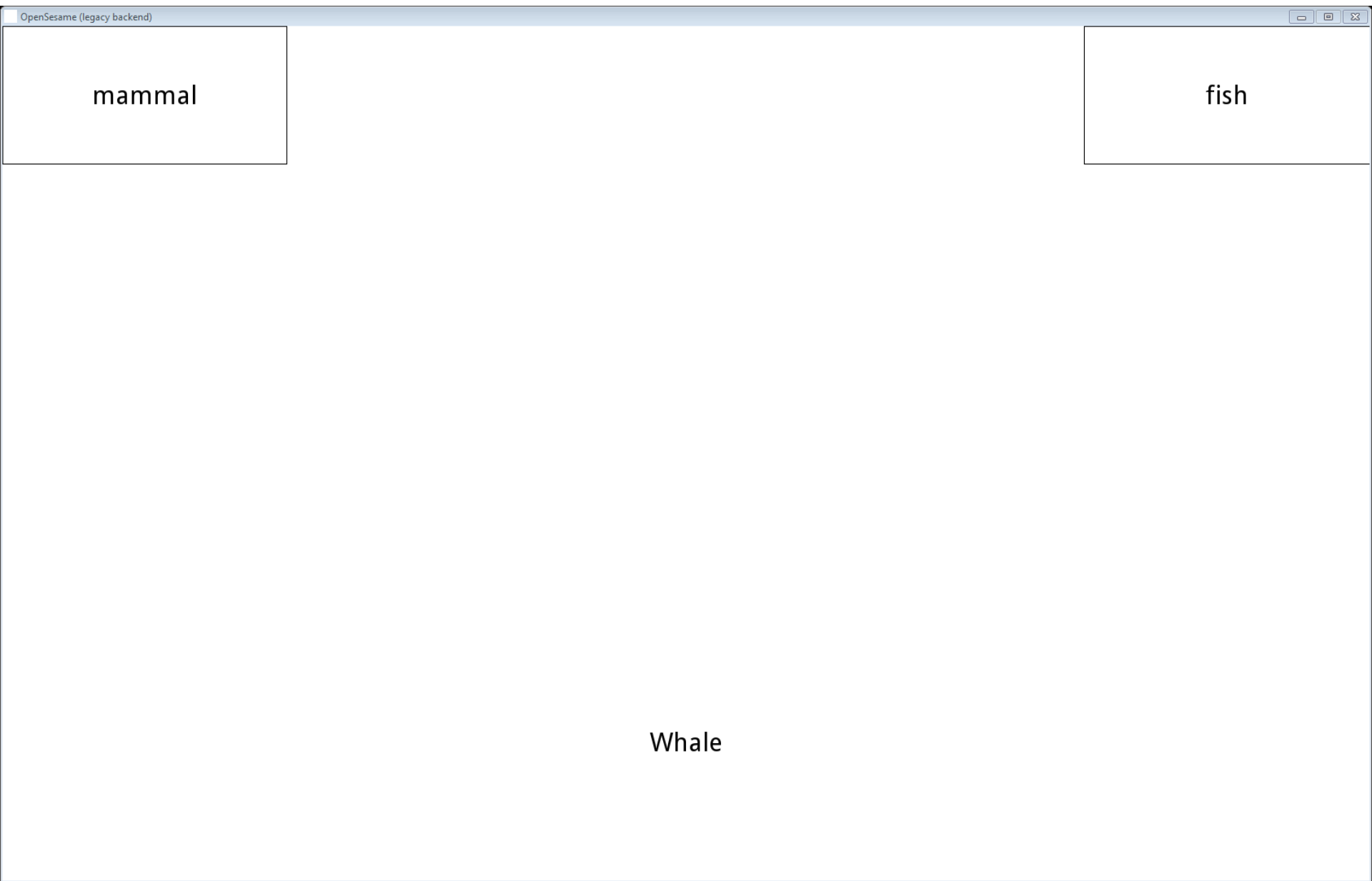
- ▣ **Typical exemplars** only share features with correct category (e.g., cat as mammal)
- ▣ **Atypical exemplars** share both features with correct and competing category (e.g., whale with mammal and fish)



### □ Main hypothesis

- ▣ **Increased competition** when categorizing atypical exemplars
- Mouse trajectories with deviation towards competing category





# OpenSesame

## Overview

73

- Graphical experiment builder
  - ▣ Developed by Sebastiaan Mathôt (Mathôt, Schreij, & Theeuwes, 2012)
  - ▣ Create experiments by **drag & drop** via GUI
  - ▣ Implement complex tasks using **Python** scripts
- Open source & cross platform
  - ▣ Download and documentation: <http://osdoc.cogsci.nl/>
  - ▣ Available for Windows, Linux, Mac OS
- Allows for extensions via plugins
  - ▣ **PyGaze** plugin for eye-tracking (Dalmaijer, Mathôt, & Stigchel, 2014)
  - ▣ **Psynteract** plugin for interactive experiments (Henninger, Kieslich, & Hilbig, 2017)
  - ▣ **Mousetrap** plugin for mouse-tracking (Kieslich & Henninger, 2017)



# OpenSesame

## Resources

74

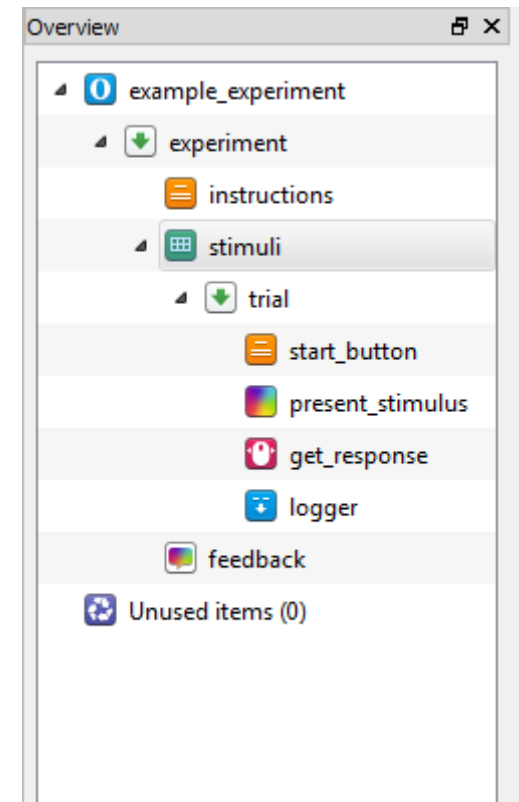
- Documentation
  - ▣ <http://osdoc.cogsci.nl>
  - ▣ Extensive & comprehensible
- Forum
  - ▣ <http://forum.cogsci.nl>
  - ▣ Very supportive & fast responses
- Source code
  - ▣ <https://github.com/smathot/OpenSesame>
- Custom search
  - ▣ <http://osdoc.cogsci.nl>
  - ▣ searches documentation & forum

# OpenSesame

## Items

75

- **Items** as the building blocks of an experiment
- **Different types** of items serve basic purposes
  - ▣ Presentation of stimuli
  - ▣ Collection of responses
  - ▣ Logging of responses
  - ▣ ...
- Organized in **hierarchical + sequential** structure
  - ▣ **Sequence** runs multiple items in succession
  - ▣ **Loop** repeats sequence of items multiple times (with variations)



# OpenSesame

## Getting started

76

- **New:** List of available templates
  - ▣ **Default:** create an experiment from scratch
  - ▣ **Extended:** already includes a basic experimental structure
- **Recent:** List of recently opened experiments

### Get started!

Welcome to OpenSesame! How can I help you?

Start a new experiment:

Default template

Extended template

Questionnaire template

Android template

Eye-tracking template

Continue with a recent experiment:

Teil2\_final.oexp

mousetrap\_response.oexp

gmc\_task\_conditions\_v6\_post\_test.oexp

gmc\_task\_conditions\_v6.oexp

Have you considered supporting OpenSesame? It's easy and quick.

Donate through PayPal

Or learn more:

Read the documentation

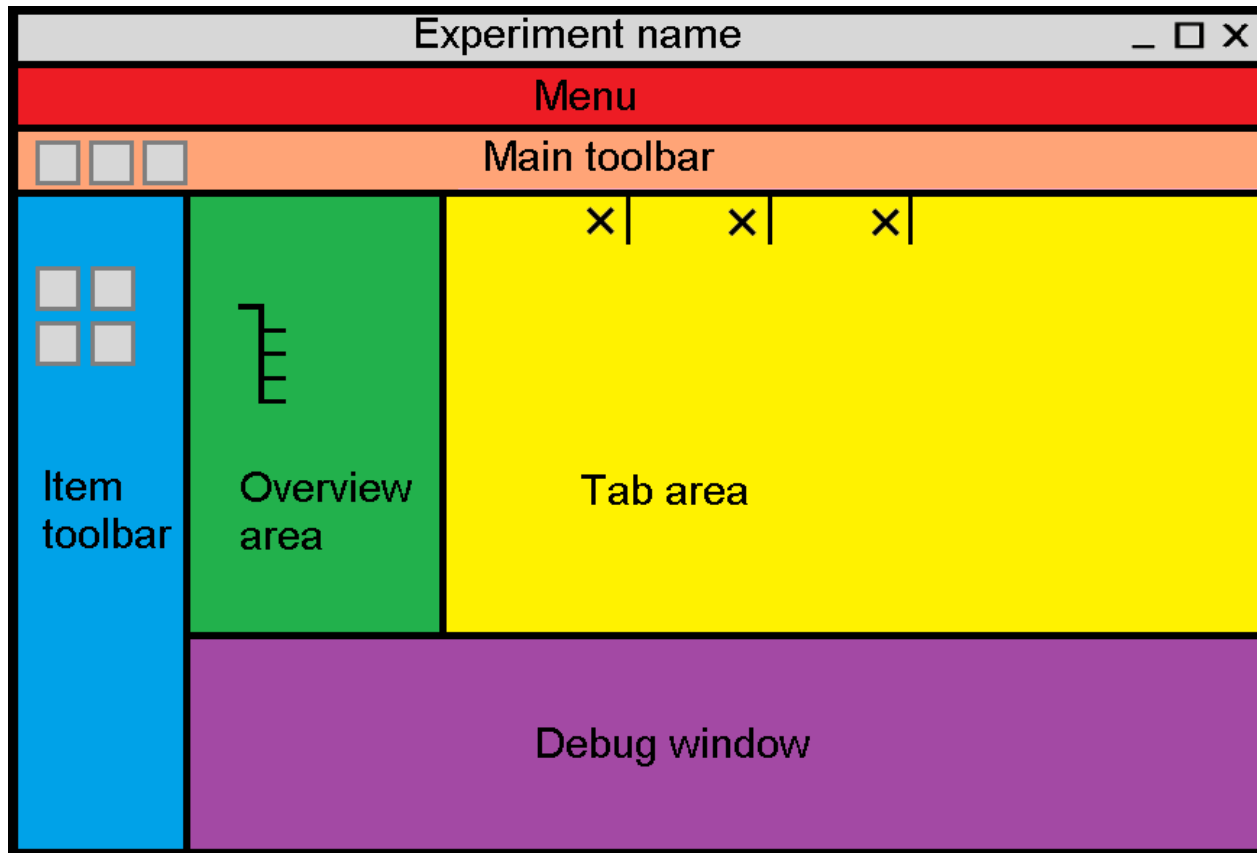
Ask a question on the forum

Dismiss this message

# OpenSesame

## Graphical user interface (GUI)

77



► <http://osdoc.cogsci.nl/manual/interface/>

# OpenSesame

## General experiment properties

78

- Set **general properties** for the experiment
  - Set the **experiment backend** to legacy or expyriment
  - Change the **name** of the experiment
  - **Foreground** color (= default font color) → black
  - **Background** → white
  - **Font** family → sans
  - **Resolution** → adjust to display resolution (mouse-tracking experiments are typically run full-screen) → **use: 1280 x 1024 px** (to match lab computers)

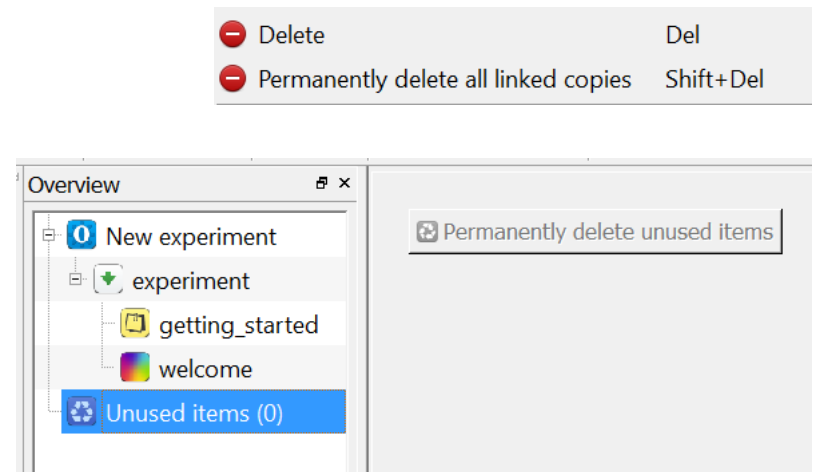
The screenshot shows the 'New experiment' dialog box in OpenSesame. The title is 'New experiment – experiment' with a subtitle 'The main experiment item'. The 'Back-end' is set to 'xpyriment [uses Expyriment]' with a dropdown arrow. Below it is a 'Back-end settings' button and a link 'Why is this important?'. The 'Resolution' is set to '1024 px' x '768 px'. The 'Foreground' color is 'black' and the 'Background' color is 'white', each with a color selection icon. The 'Font' is set to 'sans' with a dropdown arrow, and the size is '18 px'. There are checkboxes for 'Italic' and 'Bold'. An 'Example' button is to the right of the font settings. Under 'Miscellaneous', there are checkboxes for 'Uniform coordinates' (checked) and 'Bi-directional-text support'. At the bottom is a 'General script' button.

# OpenSesame

## Deleting items

79

- Delete items
  - ▣ By right clicking on them and selecting the corresponding option
  - ▣ By pressing delete (“Del”/“Entf”)
- **Task:** Delete starting items
- Deleted items still available & might still affect experiment
  - ▣ Can be re-added (by dragging them back to the experiment)
  - ▣ Permanently delete them by selecting corresponding option in unused items tab → usually do this





# OpenSesame

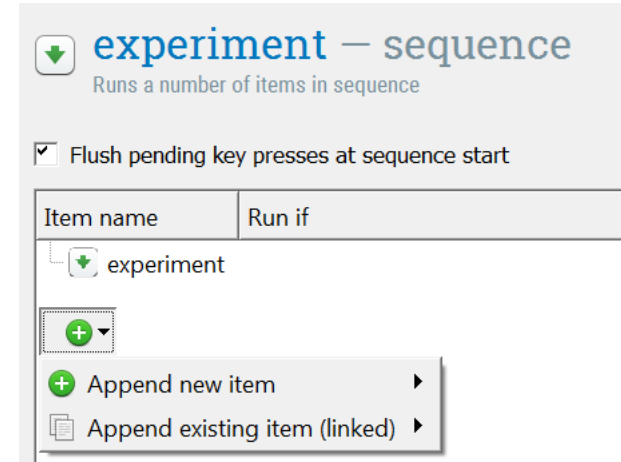
## Adding new items

80

- Two ways to add items
  - ▣ Drag them from the item toolbar onto overview area
  - ▣ Click on the relevant sequence and select append new item (or existing item)

### ► Task

- ▣ Create Welcome screen (sketchpad)
- ▣ Should contain welcome message
- ▣ Rename it *welcome*
- ▣ Should be displayed for 3000 ms

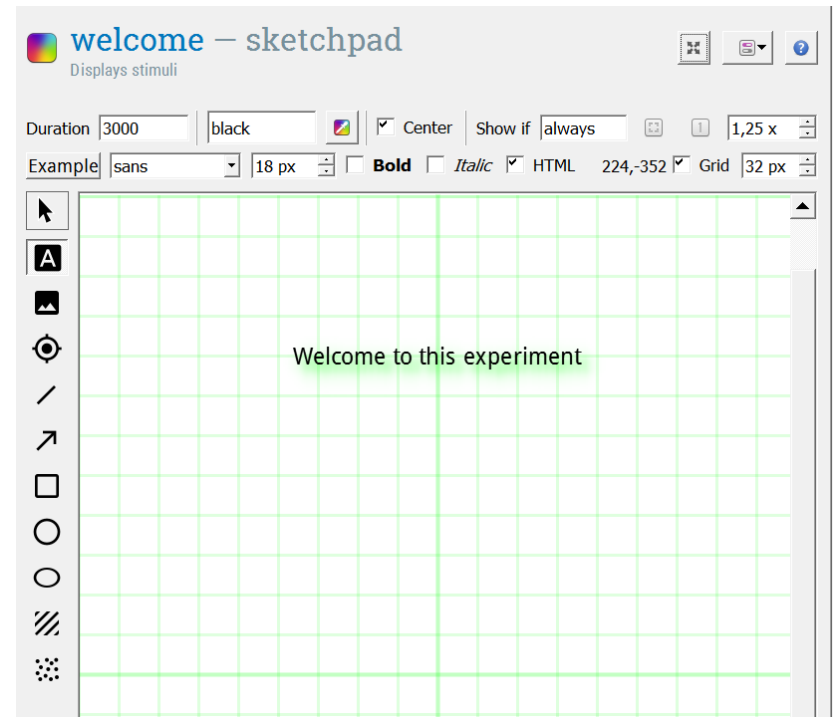


# OpenSesame

## Sketchpads

81

- Used to present strings of text, geometric shapes or bitmap images
- Provides simple built-in drawing tools
- Uses a coordinate system (center coordinate 0,0 and pixel metric)



► <http://osdoc.cogsci.nl/manual/stimuli/visual/>

# OpenSesame

## Running your experiment












82

- ▶ Run fullscreen
- ▶▶ Run in window
- ▶▶▶ Test run (run in window without specifying subject number & file location)
- Abort running experiment: press escape key
- After experiment is finished
  - ▣ Log file can additionally be saved within the experiments file pool
  - ▣ Usually we don't want this, especially not for test runs
- Task
  - ▣ Give your experiment a test run

# OpenSesame

## Sketchpads: Drawing tools


83

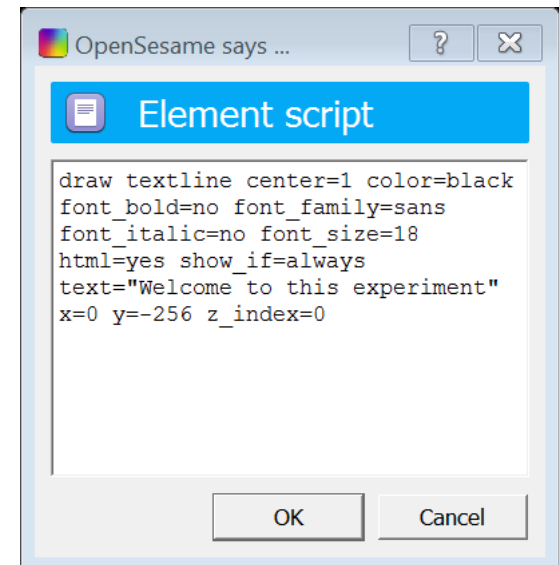
-  Select and modify (modify & move existing elements around)
-  Text element (line breaks are possible)
-  Image element
-  Fixation dot element
-  Line element
-  Arrow element
-  Rectangle element
-  Circle element
-  Ellipse element
-  Gabor patch element
-  Noise patch element

# OpenSesame

## Sketchpad: Modifying elements on sketchpad

84

- Select modify 
- Move elements: left click on them and drag them
- Modify elements: different options
  - ▣ Left click on them and change the options above
  - ▣ Double left click on them (especially to edit text)
  - ▣ Right click on them, select “edit script” and modify the underlying OpenSesame script syntax
- Delete elements: left click on them and press delete (“Entf”)
- Task
  - ▣ Change appearance of welcome message: it should be displayed in **blue and boldface**

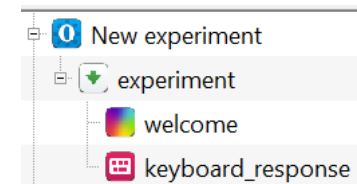
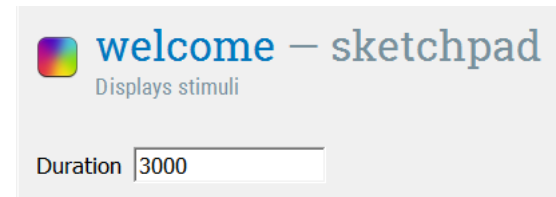


# OpenSesame

## Display duration & input of sketchpad/feedbacks

85

- Values for duration
  - ▣ Any positive integer: display duration in ms
  - ▣ *keypress*: display until any key is pressed
  - ▣ *mouseclick*: display until any mouse button is clicked
- For more specific keypress/mouseclick conditions
  - ▣ Set duration to 0
  - ▣ Use keyboard\_response/mouse\_response items

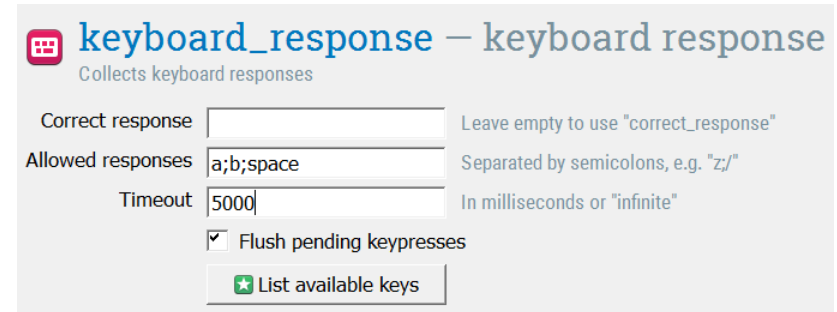


# OpenSesame

## Keyboard response

86

- Separate different keys by ;
- **Flush** pending keypresses
  - ▣ always a good idea
  - ▣ otherwise unprocessed keypresses from previous stage of experiment may exert an unwanted influence
- **Special** key names
  - ▣ See list available keys
- Task
  - ▣ Set duration of sketchpad to 0
  - ▣ Participants can press space bar to continue (use keyboard\_response item)



The screenshot shows the configuration window for the 'keyboard\_response' item in OpenSesame. The title bar reads 'keyboard\_response — keyboard response' with a subtitle 'Collects keyboard responses'. The interface includes several input fields and a checkbox:

- Correct response:** An empty text field with a hint: 'Leave empty to use "correct\_response"'.
- Allowed responses:** A text field containing 'a;b;space' with a hint: 'Separated by semicolons, e.g. "z;/"'.
- Timeout:** A text field containing '5000' with a hint: 'In milliseconds or "infinite"'.
- Flush pending keypresses:** A checked checkbox.
- List available keys:** A button with a green plus icon.

# Creating mouse-tracking experiments

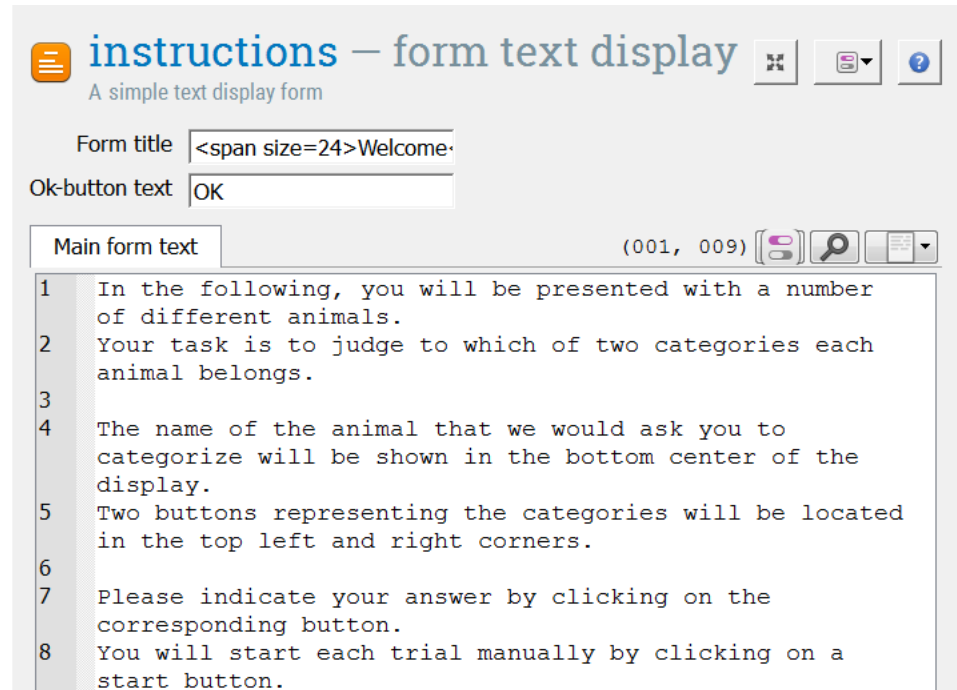
## Instruction screen

87

- Add **form\_text\_display** item to provide basic **instructions**

- Save experiment 

- Give it a **test run** 



The screenshot shows a software window titled "instructions — form text display". Below the title bar, there is a subtitle "A simple text display form". The window contains two input fields: "Form title" with the value "<span size=24>Welcome" and "Ok-button text" with the value "OK". Below these fields is a tab labeled "Main form text". To the right of the tab is a status bar showing "(001, 009)" and three icons. The main text area contains a list of instructions numbered 1 through 8:

- 1 In the following, you will be presented with a number of different animals.
- 2 Your task is to judge to which of two categories each animal belongs.
- 3
- 4 The name of the animal that we would ask you to categorize will be shown in the bottom center of the display.
- 5 Two buttons representing the categories will be located in the top left and right corners.
- 6
- 7 Please indicate your answer by clicking on the corresponding button.
- 8 You will start each trial manually by clicking on a start button.



# Creating mouse-tracking experiments

## Mousetrap plugin for OpenSesame

88

- Mousetrap plugin for OpenSesame
  - ▣ Enables users to implement mouse-tracking via **graphical user interface**
  - ▣ Also provides **Python classes** for mouse-tracking in Python code
  
- Installation (of current development version)
  - ▣ See <https://github.com/pascalkieslich/mousetrap-os#installation>
  - ▣ Execute following commands in OpenSesame's debug window

```
import pip
pip.main(['install', 'https://github.com/PascalKieslich/mousetrap-os/archive/master.zip'])
```
  - ▣ OpenSesame needs to be run in admin mode for this (under Windows)
  - ▣ Restart OpenSesame afterwards

# Creating mouse-tracking experiments

## Mousetrap plugin for OpenSesame

89

### □ Resources

- Documentation and example experiments:  
<https://github.com/pascalkieslich/mousetrap-os>
- Article (including tutorial and validation):  
Kieslich, P. J., & Henninger, F. (2017). Mousetrap: An integrated, open-source mouse-tracking package. *Behavior Research Methods*, 49(5), 1652-1667.  
<https://doi.org/10.3758/s13428-017-0900-z>

### □ Questions and updates

- Forum for questions: <http://forum.cogsci.nl/index.php?p=/categories/mousetrap>
- Mailing list for updates: <http://eepurl.com/co1AqX>

# Creating mouse-tracking experiments

## Mousetrap plugin for OpenSesame

90

- Two different options for implementing mouse-tracking



### **mousetrap\_response item**

- Tracks mouse movements while stimulus display is provided by another item
- Can use graphical user interface for designing the stimulus display



### **mousetrap\_form item**

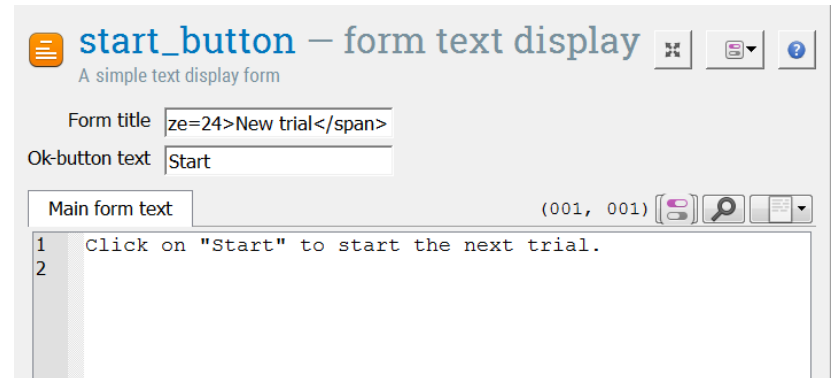
- Provides both stimulus display and mouse-tracking
- Stimulus designed using simple OpenSesame script syntax

# Creating mouse-tracking experiments

## Start button

91

- Add **form\_text\_display** item that contains the start button
- Participants start stimulus presentation by clicking on this button
- This ensures that **start position** of cursor is **comparable** across trials
- This start position usually is in the **bottom center** of the screen

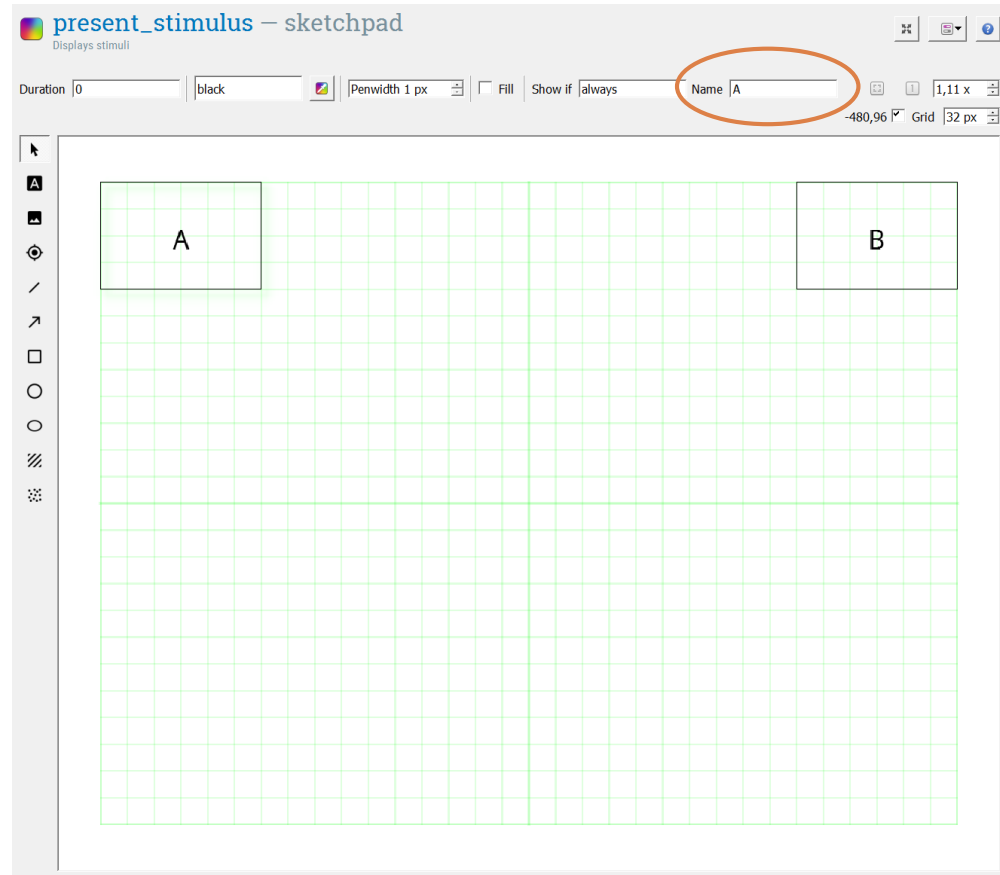


# Creating mouse-tracking experiments

## Stimulus display

92

- Use a **sketchpad** item to create the stimulus display
- Create two **buttons**
  - Draw their borders using **rect elements** and give them a label via the **Name field**
  - Add the button text using **textline elements**
- Layout considerations
  - Place buttons in **screen corners** to avoid overshooting
  - Ensure **symmetric layout** and that buttons have same distance from start position (bottom center)

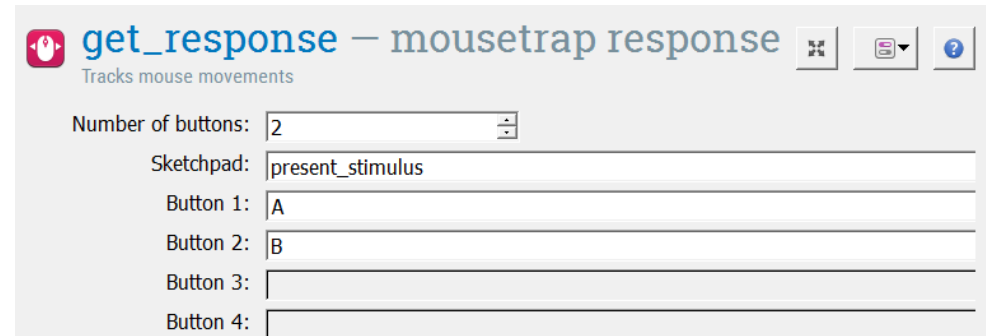
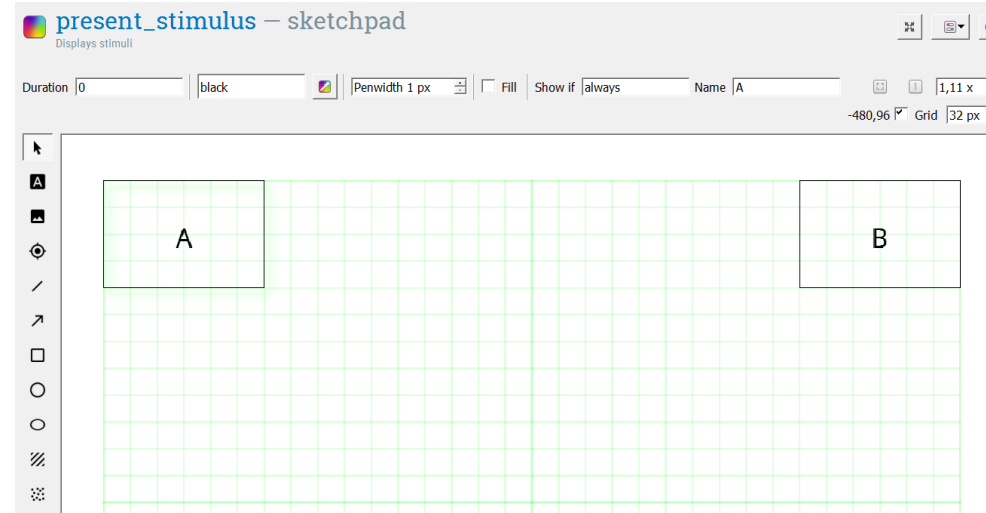


# Creating mouse-tracking experiments

## Track mouse movements

93

- Set duration of sketchpad to 0
- Add a **mousetrap\_response** item to track mouse movements and specify **number** of buttons, their **sketchpad** and **name**
- As name we enter the **label text** – this will be saved when button is clicked
- Run experiment and check data using **variable inspector**



# Creating mouse-tracking experiments

## Trial structure

94

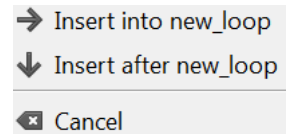
- Loops and sequences are used to structure the experiment

### Sequences

- ▣ List of items that is executed sequentially
- ▣ Does not repeat automatically – need to combine it with a loop

### Loops

- ▣ Repeatedly calls a single other item – usually a sequence (drag new sequence item on loop or select existing sequence)
- ▣ In the loop item, variables can be stored that vary for each trial
- ▣ Thereby, independent variables can be varied and/or experimental material can be stored for the experiment



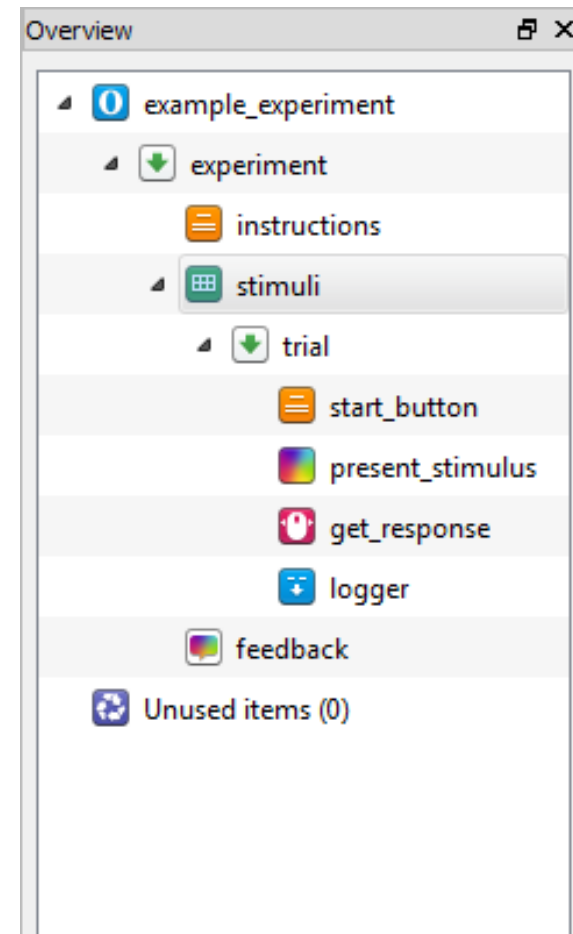
► <http://osdoc.cogsci.nl/manual/structure/loop/>

# Creating mouse-tracking experiments

## Trial structure

95

- Implement trial structure by including a **stimuli loop** and a **trial sequence**
- Include all items except the instructions in the trial sequence as they should be presented in each trial
- Add a **logger item** at the end of the trial sequence which writes the recorded data to the logfile





# Creating mouse-tracking experiments

## Loops

96

- Loops can be used to store material or to vary independent variables
  - ▣ **Variables** stored in columns
  - ▣ Distinct **trials/conditions** stored in rows (= cycles)
- **Repeat**: number of repetitions of each cycle
  - ▣ Total repetitions = cycles x repeat
  - ▣ If repeat is < 1, a randomly selected subset of trials is chosen (e.g., 50 % of trials for 0.5)
- **Order** refers to the order with which all trials are presented
  - ▣ **Random** = every trial is randomly drawn without replacement (ignoring cycles x repeat structure)
  - ▣ **Sequential** = every trial in sequential order

**block\_loop – loop**  
Repeatedly runs another item

Run ▼ trial\_sequence Break if never

Repeat each cycle 1,00 x ☒ Evaluate on first cycle

Order random ☐ Resume after break

Source table ★ Full-factorial design

Preview

Summary: **trial\_sequence** will be called **8** times in **random** order. The number of rows

	Number				
1	1				
2	2				
3	3				
4	4				
5	6				
6	7				
7	8				
8	9				


► <http://osdoc.cogsci.nl/manual/structure/loop/>


# Creating mouse-tracking experiments

## Material loop

97

- Design options of the loop are displayed at the top
- Material can be added in the bottom
  - ▣ each **column** corresponds to one experimental **variable**
  - ▣ each **row** corresponds to one unique **trial**
- Create the loop as displayed on the right


 **stimuli – loop**  
Repeatedly runs another item

Run  trial

Break if


Repeat

☒ Evaluate on first cycle

Order  random

☐ Resume after break

Source

 Full-factorial design

Preview

Summary: **trial** will be called **4 times** in **random** order. The number of rows is 4. All rows occur once.

	Exemplar	CategoryLeft	CategoryRight	CategoryCorrect	Condition
1	Cat	mammal	reptile	mammal	Typical
2	Hawk	reptile	bird	bird	Typical
3	Penguin	bird	fish	bird	Atypical
4	Whale	fish	mammal	mammal	Atypical

# Creating mouse-tracking experiments

## Choosing variable names

98

- OpenSesame (and Python) is **case sensitive**
  - ▣ “Condition” and “condition” are different variables
- Variable names **may not contain spaces**
  - ▣ use “\_” instead, e.g., “my\_variable”
- Variables may **not contain . or ,**

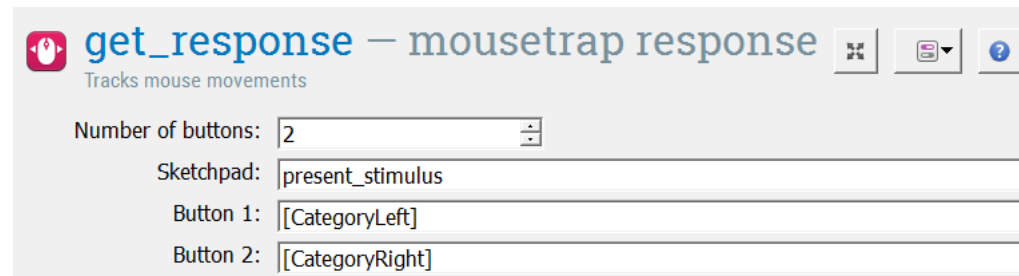
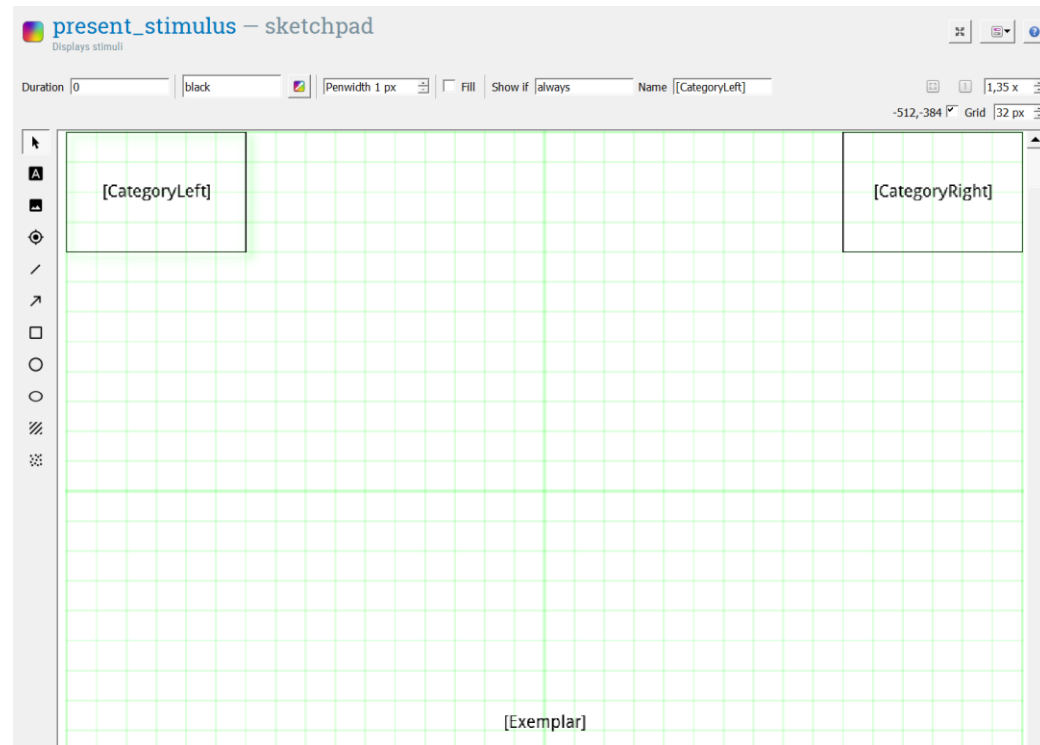
► <http://osdoc.cogsci.nl/manual/variables/>

# Creating mouse-tracking experiments

## Using variables

99

- Variables that are defined in a loop or recorded through an item can be used by including their name in **square brackets**
- **Adjust the button labels** in sketchpad and mousetrap\_response item so that they vary in each trial
- Present **exemplar** name in bottom center of screen
- Give the experiment a test run



# Creating mouse-tracking experiments

## Built-in experimental variables (excerpt)

100

### □ Experiment variables

- ▣ `subject_nr` (1, 2, ...) & `subject_parity` (“odd” or “even”)

### □ Item variables

- ▣ `count_item_name`: number of times – 1 **item** has been called (starting at 0)
- ▣ `time_item_name`: timestamp of last time item was executed

### □ Response variables

- ▣ `response` & `response_item_name` → last response overall and to specific **item**
- ▣ `response_time` & `response_time_item_name` → response time in ms
- ▣ `correct` & `correct_item_name` → =1 if `response == correct_response`, 0 otherwise

### □ Feedback variables

- ▣ `average_response_time/avg_rt`: average response time
- ▣ `accuracy /acc`: average percentage of correct responses
- ▣ can be reset, e.g., are reset by default when using feedback item → take care!

► <http://osdoc.cogsci.nl/manual/variables/#built-in-variables>

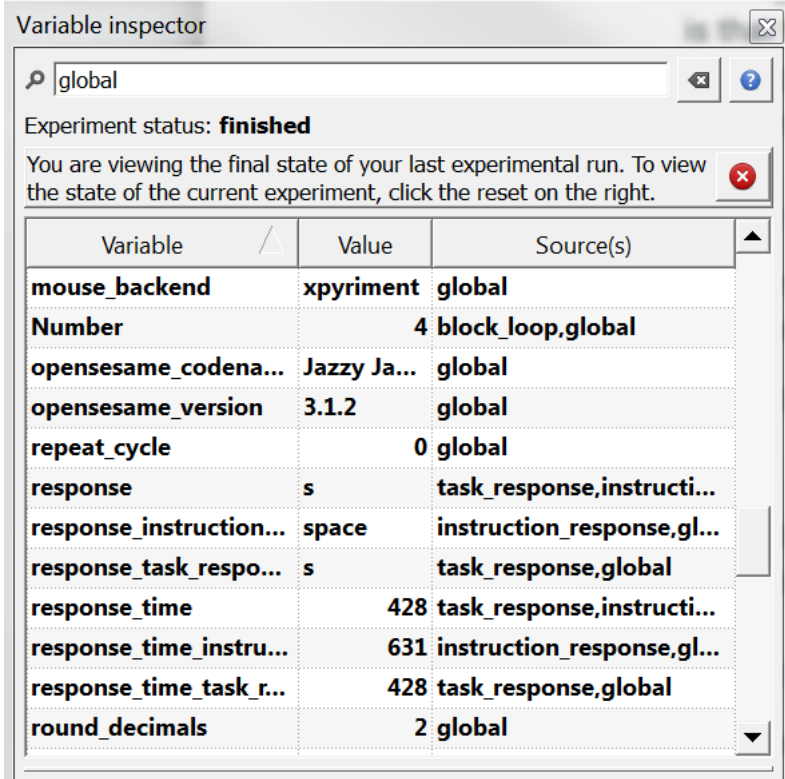
# Creating mouse-tracking experiments

## Variable inspector

101

### Overview of experimental variables: **variable inspector**

- User defined variables  
(e.g., via loop or inline\_scripts)
- Built-in variables
- Displays the current status of all experimental variables
  - Can be used to monitor changes while experiment is running
  - For this the Runner (specified in Tools/Preferences) cannot be set to opensamerun



Variable inspector

global

Experiment status: **finished**

You are viewing the final state of your last experimental run. To view the state of the current experiment, click the reset on the right.

Variable	Value	Source(s)
mouse_backend	xpyriment	global
Number	4	block_loop,global
opensesame_codena...	Jazzy Ja...	global
opensesame_version	3.1.2	global
repeat_cycle	0	global
response	s	task_response,instructi...
response_instruction...	space	instruction_response,gl...
response_task_respo...	s	task_response,global
response_time	428	task_response,instructi...
response_time_instru...	631	instruction_response,gl...
response_time_task_r...	428	task_response,global
round_decimals	2	global

► <http://osdoc.cogsci.nl/manual/variables/#the-variable-inspector>

# Creating mouse-tracking experiments

## Response collection & logging

102



Variables only saved in the log file **if there is a logger item!**

- ▣ Every time a logger item is passed, it writes the current status of all experimental variables to the log file
- ▣ Always check that you have a logger item included in your trial
- ▣ Always use the **same logger** by using append existing item / linked copy
- ▣ Always check your log file

▣ **Default** (log all variables) makes sense

- ▣ In case a variable is not logged, add it as a custom variable but inspect experiment closely, as there might be a problem somewhere

logger — logger  
Logs experimental data

☒ Log all variables (recommended)

	Custom variable	Source(s)

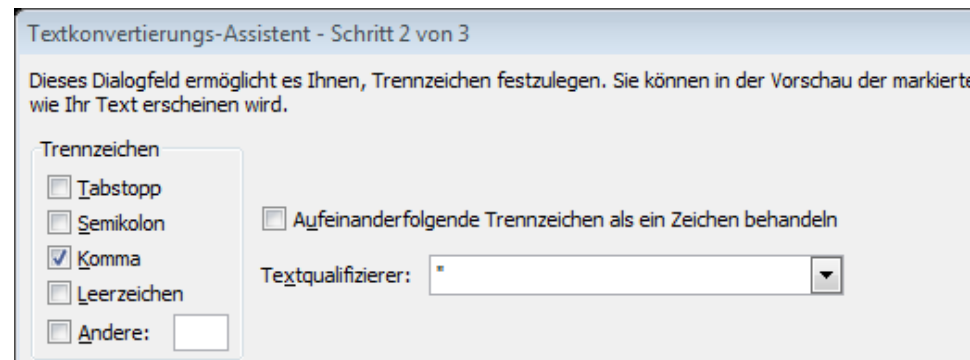
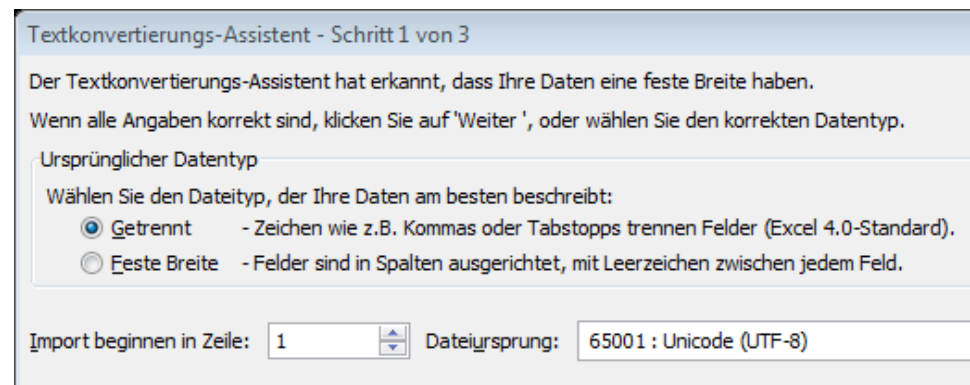
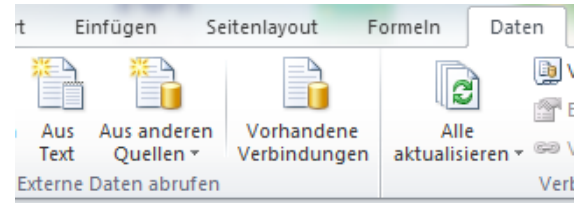
► <http://osdoc.cogsci.nl/manual/logging/>

# Creating mouse-tracking experiments

## Opening logfiles

103

- Logfiles are saved in csv format – specifications
  - ▣ Separator: “,”
  - ▣ Decimal point (not decimal comma)
  - ▣ Encoding: UTF-8
- Open csvs in Excel via text import assistant
- Task
  - ▣ Run experiment once and look at the logfile





# Creating mouse-tracking experiments

## Copying items

104

### □ Different options for copying an item

#### ▣ Unlinked copy

- create a duplicate of the item that is not connected to the original item anymore
- → if original item is changed, new item is not affected

#### ▣ Linked copy

- create a duplicate of the item that is actually the very same item
- → if either item is modified, the other changes as well

	Open	
	Rename	F2
	Copy (unlinked)	Ctrl+C
	Copy (linked)	Ctrl+Shift+C
	Delete	Del
	Permanently delete all linked copies	Shift+Del
	Help	

# Creating mouse-tracking experiments

## Conditional presentation of items and elements

105



**Run if** in sequence and **show if** on sketchpad

- Specify conditions under which an item is presented
- Specify complex conditions by using **ands** and **ors**

Can be used

- ... for implementing experimental conditions
- ... for giving feedback
- ... for adapting to a participant response

Examples

- `[correct]==1`
- `[correct]==1 and [response_time] >= 2000`
- `[condition]==2 or [condition]==3`
- `[condition]!=1`

Additional values are: always and never (for ignoring items)

► <http://osdoc.cogsci.nl/manual/variables/#using-conditional-if-statements>



**trial\_sequence** — sequence

Runs a number of items in sequence

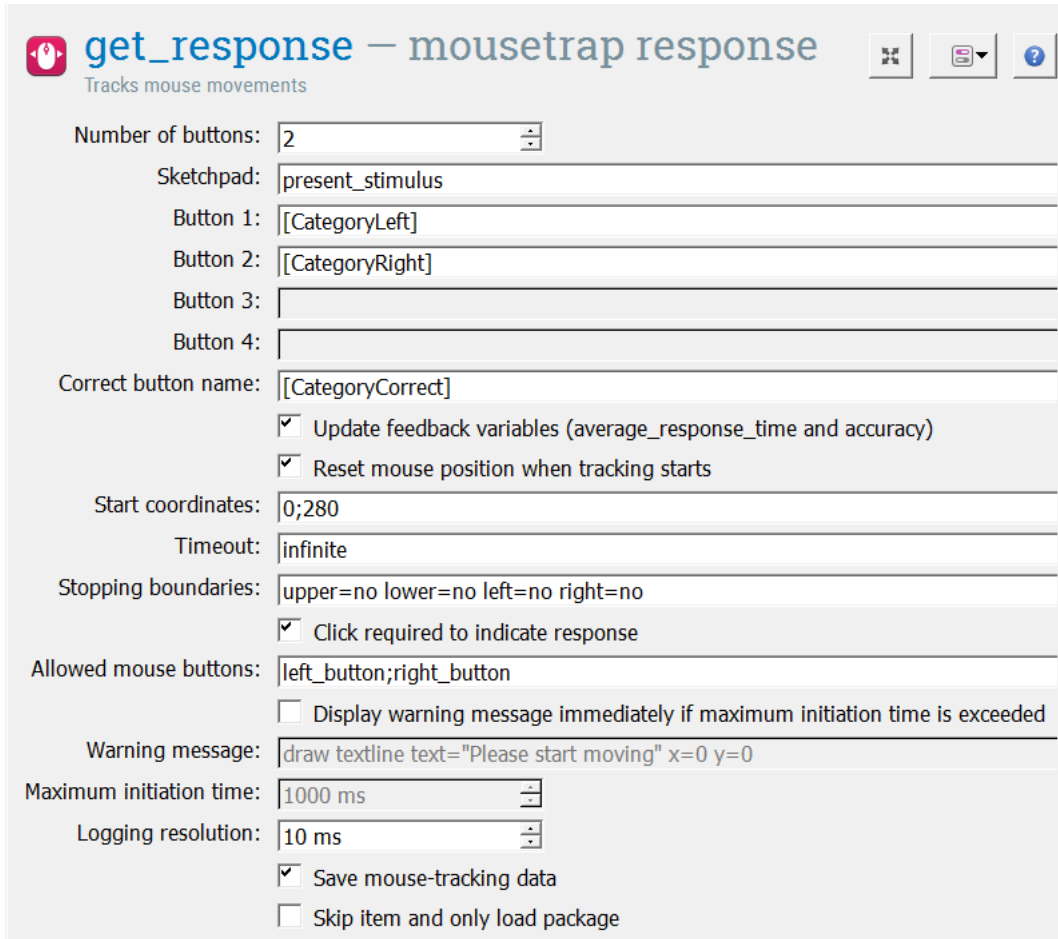
☒ Flush pending key presses at sequence start

Item name	Run if
trial_sequence	
fixation_dot	always
task_stimulus	always
task_response	always
feedback_correct	[correct]==1
feedback_wrong	[correct]==0
logger	always

# Creating mouse-tracking experiments

## Mouse-tracking settings

106



The screenshot shows the 'get\_response' mouse-trap response settings interface. The title bar reads 'get\_response — mousetrap response' with a subtitle 'Tracks mouse movements'. The interface includes several input fields and checkboxes for configuring the experiment.

Number of buttons: 2

Sketchpad: present\_stimulus

Button 1: [CategoryLeft]

Button 2: [CategoryRight]

Button 3:

Button 4:

Correct button name: [CategoryCorrect]

☒ Update feedback variables (average\_response\_time and accuracy)

☒ Reset mouse position when tracking starts

Start coordinates: 0;280

Timeout: infinite

Stopping boundaries: upper=no lower=no left=no right=no

☒ Click required to indicate response

Allowed mouse buttons: left\_button;right\_button

☐ Display warning message immediately if maximum initiation time is exceeded

Warning message: draw textline text="Please start moving" x=0 y=0

Maximum initiation time: 1000 ms

Logging resolution: 10 ms

☒ Save mouse-tracking data

☐ Skip item and only load package

- Define buttons
- Automatically **code correctness** of response (correct = 0 or 1), e.g., to provide feedback
- **Reset cursor position** at tracking onset to exact coordinates
- **Limit** maximum **response time**
- Use **dynamic start** procedure (cf. Scherbaum & Kieslich, in press)
- End trial on **click** vs. on **touch**
- Issue immediate **warning** for long **initiation times**

# Creating mouse-tracking experiments

## Mousetrap settings

107

- Task: Try out different mousetrap\_response settings
  - ▣ Center the cursor position on exact coordinates when tracking starts
  - ▣ Record response and finish tracking as soon as the cursor touches one of the buttons (needs not to be clicked)
  - ▣ Only allow clicks on the left button of the physical mouse
  - ▣ Specify which variable contains the correct response and give feedback to the participant

# Creating mouse-tracking experiments

## File pool & pictures

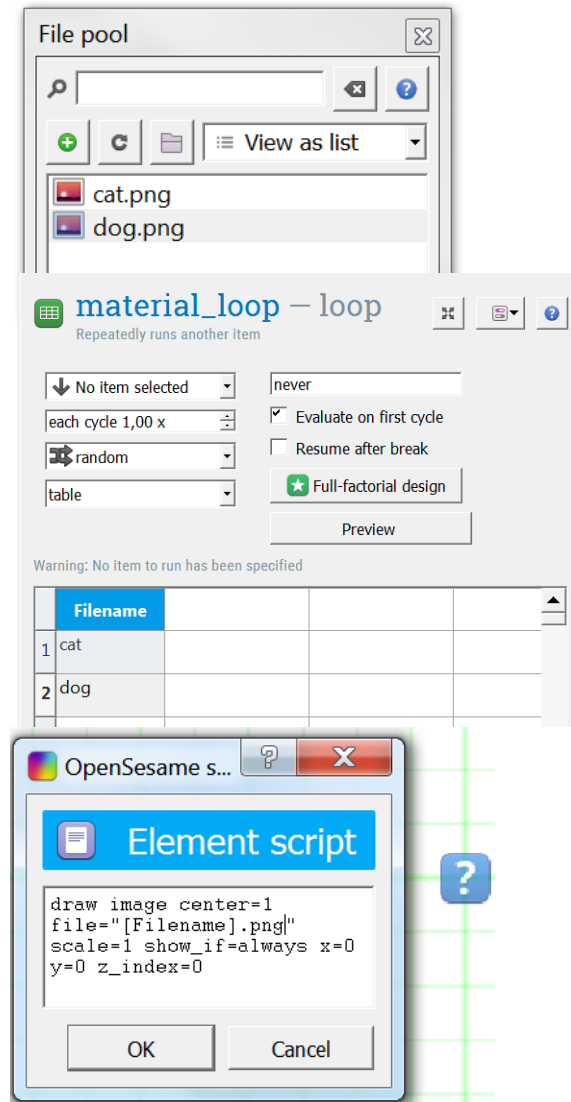
108

### File pool

- Is included in the experiment file itself
- Can be used to include pictures (and other files) in the experiment

### Pictures

- Can have various formats – however, especially the **.png** format seems recommendable
- Rescaling should not be performed in OpenSesame but before, as it can lead to lags between trials
- If pictures should vary between trials, set the file name in a loop and change the OpenSesame script of the image





# Randomization & manipulations

# Randomizations & manipulations

## Loops

110

- Loops can be used to store material or to vary independent variables
  - ▣ **Variables** stored in columns
  - ▣ Distinct **trials/conditions** stored in rows (= cycles)
- **Repeat**: number of repetitions of each cycle
  - ▣ Total repetitions = cycles x repeat
  - ▣ If repeat is < 1, a randomly selected subset of trials is chosen (e.g., 50 % of trials for 0.5)
- **Order** refers to the order with which all trials are presented
  - ▣ **Random** = every trial is randomly drawn without replacement (ignoring cycles x repeat structure)
  - ▣ **Sequential** = every trial in sequential order

**block\_loop – loop**  
Repeatedly runs another item

Run ▼ trial\_sequence Break if never

Repeat each cycle 1,00 x ☒ Evaluate on first cycle

Order random ☐ Resume after break

Source table ★ Full-factorial design

Preview

Summary: **trial\_sequence** will be called **8** times in **random** order. The number of rows

	Number				
1	1				
2	2				
3	3				
4	4				
5	6				
6	7				
7	8				
8	9				

► <http://osdoc.cogsci.nl/manual/structure/loop/>

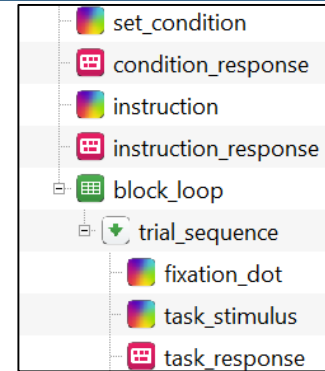
# Randomizations & manipulations

## Implementing randomizations in OpenSesame

111

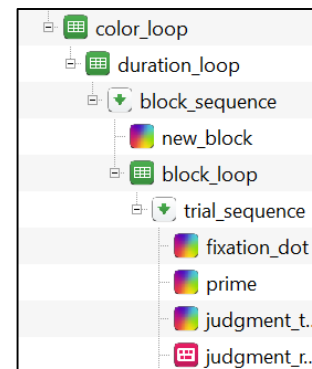
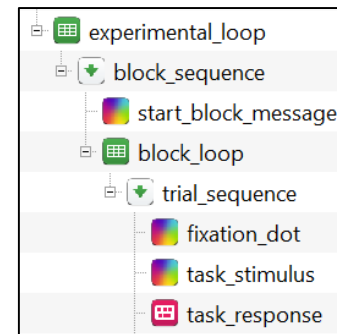
### Between participants

- i.e., participant is in one condition set for whole experiment and every participant works on the same set of items
- Assign condition at the beginning of the experiment (using a keyboard\_response item)



### Within participants

- a) **One factor** is manipulated and **same items are presented for each factor level (blocked)**
  - use one loop item for factor & nested within this a loop for material
- b) **Several factors** are manipulated and same material is presented for each combination of factor levels (**blocked**)
  - i. If all combination of factor levels should be presented **in random order**, create combinations of all levels within one loop (see (a))
  - ii. If structure is **hierarchical**, i.e., for each level of one factor first all levels of the other factor are presented: use nested loops



- Note: if items should repeatedly be presented: make use of **repeat**

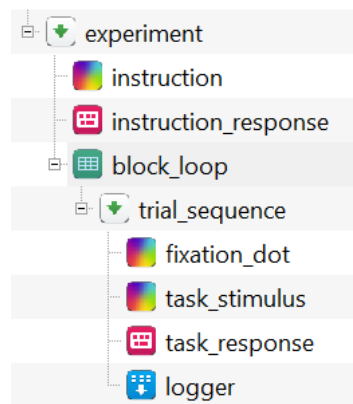


# Randomizations & manipulations

## Within participants: Full-factorial

112

- Case: **all items are presented for all factor levels**
  - Idea: One factor is manipulated within participants and same items are presented for each factor level – however, the presentation should not be blocked for factor levels but **everything is presented in random order**
  - Solution: Create one large loop where each factor level and item are combined (make use of **full-factorial design**)



Generate full-factorial design

To generate a full-factorial design, enter variable names (factors) in the first row of the table below. Below the variable names, enter the levels of the variables.

**Note:** This will overwrite the current loop table

☐ Show example

	Number	NumberSide
1	1	left
2	2	right
3	3	
4	4	
5	5	
6	6	
7	7	
8	8	

OK Cancel

block\_loop – loop

Repeatedly runs another item

trial\_sequence never

each cycle 1,00 x

random

table

☒ Evaluate on first cycle

☐ Resume after break

☒ Full-factorial design

Preview

Summary: trial\_sequence will be called 16 times in random order. number of rows is 16. All rows occur once.

	Number	NumberSide
1	1	left
2	2	left
3	3	left
4	4	left
5	6	left
6	7	left
7	8	left
8	9	left
9	1	right
10	2	right
11	3	right
12	4	right
13	6	right
14	7	right
15	8	right
16	9	right

# Randomizations & manipulations

## Crossed randomizations

113

### □ Crossed randomization

- Idea: One factor is manipulated within participants and **items** are **randomly assigned to one of the factor levels** – importantly, each item is only presented **once**
- Besides, presentation is not blocked but all items are presented in random order

### □ Solution with **nested loops**

- One loop for items – within this loop: another loop for within factor where repeat is set to **1/number of factor levels**
- Can also be used, e.g., to counterbalance presentation order when item pairs are presented
- Problem: cannot ensure that number of items (i.e., number of trials) for each factor level are exactly the same

### □ Solution using **advanced loop operations**

- Include items and within factor levels in the same loop in different columns
- Use the advanced loop operation **shuffle** to only shuffle the column containing the within factor levels

# Randomizations & manipulations

## Advanced loop operations

114

### □ Advanced loop operations

- ▣ Can be used for implementing advanced randomizations in loops
- ▣ Have to be specified in the OpenSesame script of the loop item – after the setcycle commands (i.e. at the end of the script)

### □ Particularly useful operations

- ▣ **shuffle cue**: Shuffle the column called cue → implement crossed randomizations
- ▣ **shuffle\_horiz word1 word2**: Shuffle the columns word1 and word2 horizontally → **useful for counterbalancing presentation order**
- ▣ **weight w**: Repeat each row by a weighting factor specified in column w → useful if each stimulus should be presented with a different frequency

### □ To test if a operation works as expected, use the **preview** feature

▶ <http://osdoc.cogsci.nl/manual/structure/loop/#advanced-loop-operations>

# Randomizations & manipulations

## How to implement manipulations

115

- If manipulation affects specific task characteristic (e.g., color of font or different target letters)
  - ▣ Modify specific elements with experimental variables
  - ▣ Use **show if** condition
- If difference between conditions affects complex task changes
  - ▣ Complex layout changes
    - Create different sketchpads and use **run if** condition
  - ▣ Complex task changes
    - Use different loops/sequences and set run if condition for them
    - Identical elements within the different conditions can be realized by creating linked copies



Item name	Run if
trial_sequence	
fixation_dot	always
task_stimulus_cond1	[condition]==1
task_stimulus_cond2	[condition]==2
task_response	always
logger	always

Item name	Run if
experiment	
set_condition	always
condition_response	always
block_loop_cond1	[condition]==1
block_loop_cond2	[condition]==2

# Randomizations & manipulations

## Block & trial structure

116

- Situation: Different blocks of trials are used with the same task
  - ▣ Typically: material (i.e., items) change but task remains the same
  - ▣ e.g., practice and task block
  - ▣ Solution: use different loops that run the same sequence
  
- Situation: block of trials has a specific structure
  - ▣ e.g., first 10 test items are presented, then the actual task items follow
  - ▣ Solution: the block actually consists of multiple blocks  
→ use different loops that run the same sequence

# Randomizations & manipulations

## Events at specific trials

117

- Situation: Something happens at trial X
  - ▣ e.g., task has 6 trials and there should be a break after trial 3
  - ▣ Solution: use count\_sequence to set run if condition
  - ▣ Note: trial counter starts at 0 (i.e., count\_trial\_sequence has value 0 in first trial, so if you want to make a break after 3 trials, the counter should be equal to 2)
  - ▣ More complex events can be realized using inline\_scripts (also using the count\_sequence variable)

**trial\_sequence** — sequence  
Runs a number of items in sequence

☒ Flush pending key presses at sequence start

Item name	Run if
trial_sequence	
fixation_dot	always
task_stimulus	always
task_response	always
logger	always
break	[count_trial_sequence]==2

**break** — sketchpad  
Displays stimuli

Duration


Take a break

# Randomizations & manipulations








## Manipulation of task order

118

- Situation: The order of two tasks should be manipulated
  - ▣ Solution: Create linked copy of whole loop and set run if condition

 **experiment** — sequence  
Runs a number of items in sequence

☒ Flush pending key presses at sequence start

Item name	Run if
 experiment	
 set_condition	always
 instruction	always
 instruction_response	always
 task1	[order_condition]==task1_first
 task2	always
 task1	[order_condition]==task1_last



# Design factors



# Creating mouse-tracking experiments

## Methodological considerations

120

- General challenge when designing a mouse-tracking study
  - ▣ Movements should reflect developing **commitment** not information search  
→ **minimize** amount of **new information** after tracking onset
  - ▣ **Preferences** should not develop **before** tracking starts  
→ **critical information** should only be made available at the **last** moment
- Mouse **start positions** should be **comparable** across trials
  - ▣ Participants have to click on a **centered button** to start the trial
  - ▣ Exactly identical start positions across trials achieved by **resetting** mouse or by **computational alignment** during analysis
- **Counterbalancing positions** across trials / participants
  - ▣ Vary which option is presented on which side (left vs. right)
  - ▣ Can be done between trials or between participants (depending on study)

# Design factors

## Overview

121

- Researchers face a number of **design choices** when creating mouse-tracking experiments
  - ▣ Starting procedure (static, restricted initiation time, dynamic)
  - ▣ Cursor speed settings (velocity & acceleration)
  - ▣ Indicate response via click vs. touch
- Some authors have given **recommendations** about designing mouse-tracking studies (Fischer & Hartmann, 2014; Hehman et al., 2015)
- Empirical **validation** studies are being conducted (Scherbaum & Kieslich, in press; Kieslich et al., in preparation)

# Design factors

## Preliminary summary of findings

122

### □ Response indication

- **Click** on button leads to larger effects than **touch** – effect related to higher proportion of trials with extreme movements to non-chosen option

### □ Mouse sensitivity settings

- Did not significantly influence effect of interest in static setup – although **default** settings generally lead to more extreme curvature than **reduced mouse speed**
- Reducing mouse speed becomes relevant for dynamic start condition to ensure stimulus information can be acquired during upwards movement

### □ Starting procedure

- Restricting **maximum initiation time** led to larger effects – a **dynamic start** only influenced shape but not effect size
- However, restricting initiation times also led to largest proportion of excluded trials (and seemed to be challenging for some participants)

# Design factors

## Starting procedure

123

- Simplest setup: static start
  - Stimulus presented immediately (or after a short delay) after click on start button
  - Participants can freely decide when to initiate their movement
  - Potential risk: participants make their decision first and then initiate mouse movement
  - Studies using this procedure often still find effects but in some trials (with extremely straight trajectories) this problem might occur
- Alternative: ensure early movement initiation
  - Alternative I: Participants explicitly told to initiate movement within certain time limit (around 400-600 ms) – if they take longer, warning message is displayed after the trial
  - Alternative II: dynamic start: Participants first have to move the mouse upwards for 50 px so the stimulus is presented
  - These procedures tend to lead to larger effects and more consistent movements – alternative I sometimes hard for participants to accomplish
  - Challenge: is the task simple enough that participants can make the complete decision during the upwards movement?

# Design factors

## Cursor speed and acceleration

124

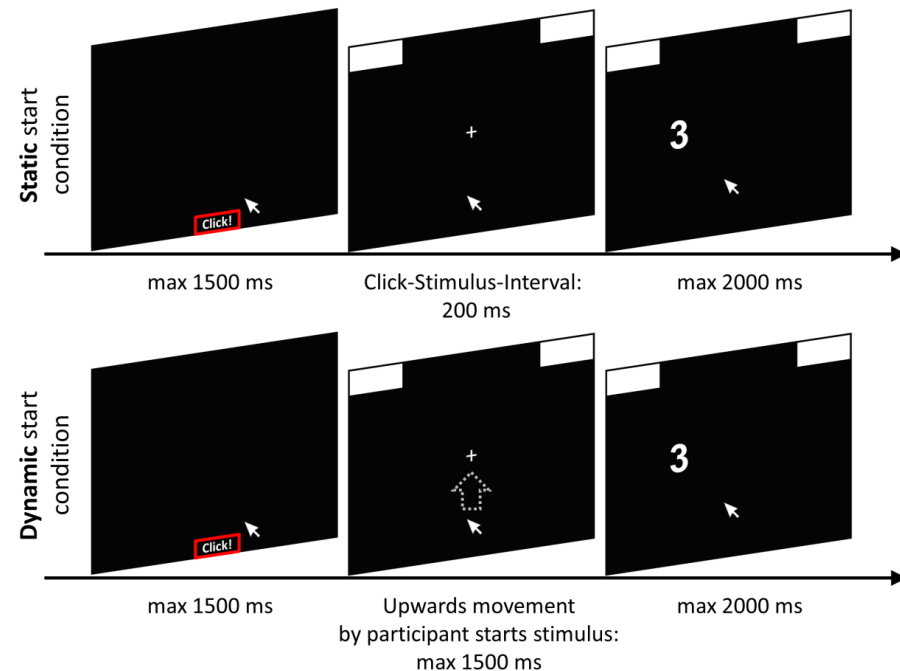
- Different settings have been employed
  - ▣ Default: medium speed, acceleration enabled
  - ▣ Slow: reduced speed, acceleration disabled
    - Reduction of speed and disabling of acceleration important for starting procedures that encourage early movement initiation → ensure that decision can be completed during upwards movement

# Starting procedure: Static vs. dynamic start

## Method (Scherbaum & Kieslich, in press)

125

- Mouse-tracking in Simon task
  - ▣ Participants **click on left vs. right** option depending on stimulus (left if number < 5, otherwise right)
  - ▣ Position of stimulus varied (left vs. right) so that desired response and position are either **congruent** or **incongruent**
- Variation starting procedure
  - ▣ **Dynamic**: move upwards to display stimulus (data from Scherbaum et al., 2010)
  - ▣ **Static**: stimulus displayed after fixed interval of 200 ms (typical duration of movement initiation in dynamic condition) (new data)

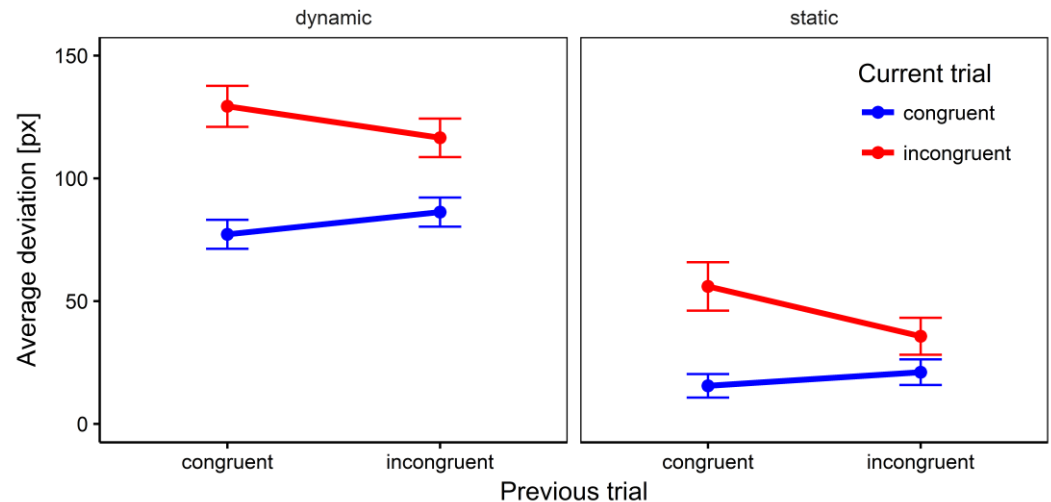


# Starting procedure: Static vs. dynamic start

## Discrete effects: Results for average deviation

126

- Simon effect and congruency sequence effect replicated in both conditions
- No significant interaction of theoretically important effects with starting procedure



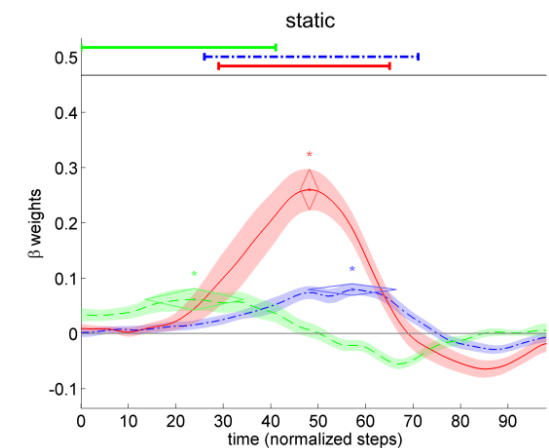
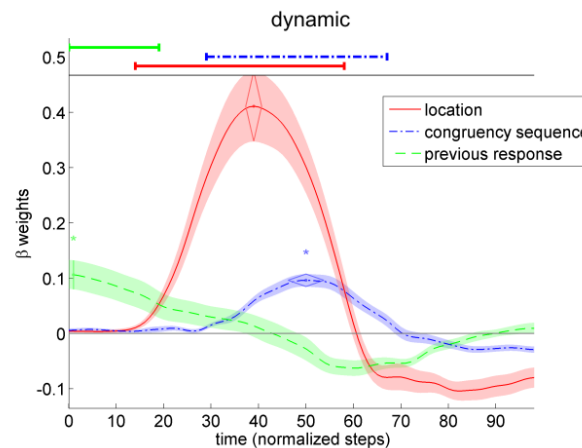
Error bars represent 1 *SEM*.

# Starting procedure: Static vs. dynamic start

## Dynamic effects: Time-continuous angle regression

127

- Time continuous multiple regression predicting vertical movement angle at each time point
- Predictors
  - ▣ location (congruency)
  - ▣ congruency sequence (same / different)
  - ▣ previous response (same / different)
- Effects stronger and more temporarily distinct in dynamic starting condition



Average  $\beta$  weights per time step and predictor.  
Lines indicate segments of  $\beta$  weights significantly  $> 0$ .

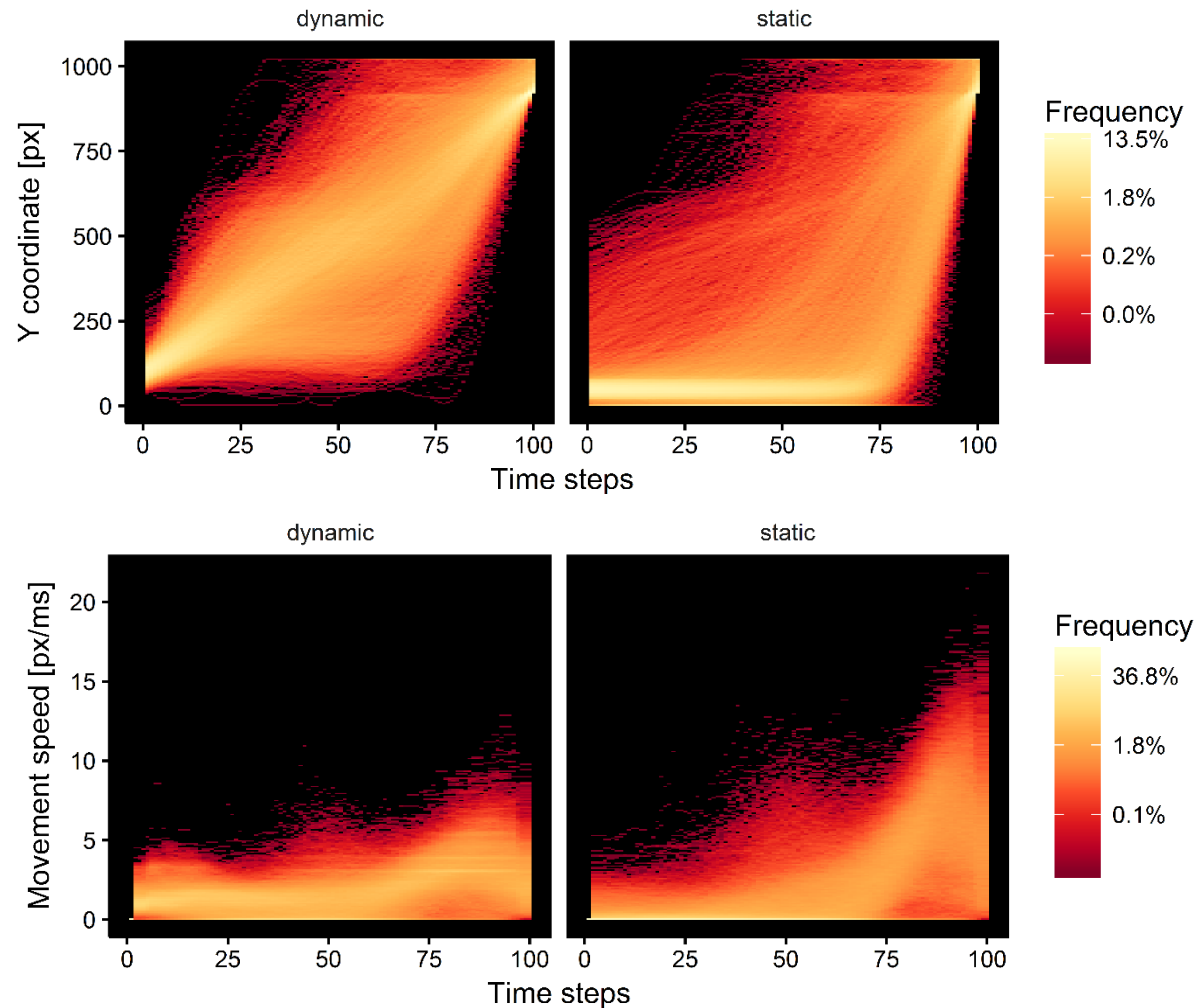


# Starting procedure: Static vs. dynamic start

## Movement consistency

128

- Smooth and consistent upwards movement in dynamic starting condition
- Participants in static starting condition often stay at bottom of screen for more than half of the trial before moving upwards quickly

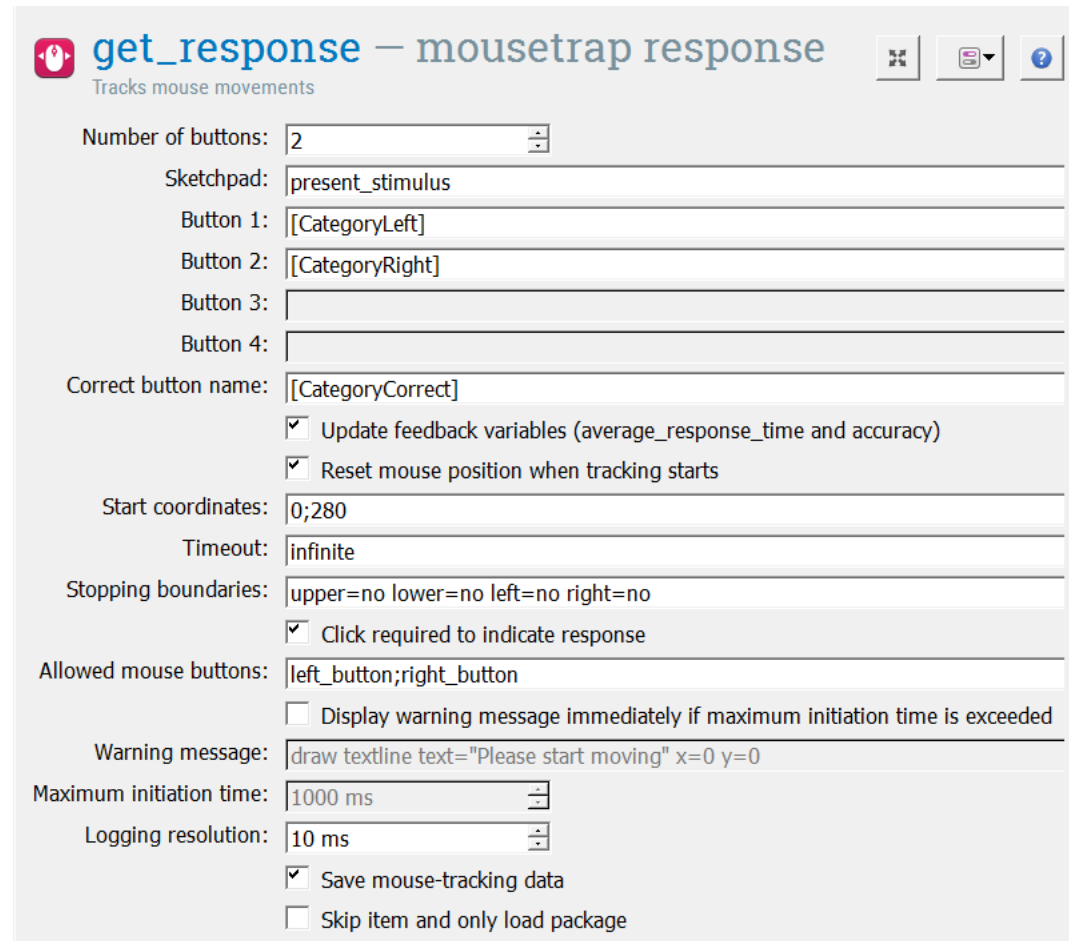


# Creating mouse-tracking experiments

## Implementing a dynamic start procedure

129

- Task: Implement a dynamic start procedure
  - ▣ Include a new sketchpad without stimulus before decision screen as well as a new mousetrap\_response item
  - ▣ Reset mouse cursor to center of the start button when tracking starts
  - ▣ Set number of buttons to 0 and specify an upper tracking boundary 50 px above start (→ subtract 50 px as px in OpenSesame increase towards the bottom)



**get\_response** — mousetrap response  
Tracks mouse movements

Number of buttons: 2

Sketchpad: present\_stimulus

Button 1: [CategoryLeft]

Button 2: [CategoryRight]

Button 3:

Button 4:

Correct button name: [CategoryCorrect]

☒ Update feedback variables (average\_response\_time and accuracy)

☒ Reset mouse position when tracking starts

Start coordinates: 0;280

Timeout: infinite

Stopping boundaries: upper=no lower=no left=no right=no

☒ Click required to indicate response

Allowed mouse buttons: left\_button;right\_button

☐ Display warning message immediately if maximum initiation time is exceeded

Warning message: draw textline text="Please start moving" x=0 y=0

Maximum initiation time: 1000 ms

Logging resolution: 10 ms

☒ Save mouse-tracking data

☐ Skip item and only load package



Your experiments

# Creating mouse-tracking experiments (Tuesday)

131

- 09:00-11:00 OpenSesame & mousetrap-os introduction
- 11:00-12:00 Build experiment
- 12:00-13:00 Lunch break
- 13:00-15:00 Build experiment
- 15:00-16:00 Register experiment at OSF (Michael)
- 16:00-17:00 Keynote Neil Stewart
- 17:00-18:00 Meet the Scientist
- 18:00-19:00 Participate in experiments (Lab computers)

# Thank you!

Questions and comments are highly appreciated!

Now & via email: [kieslich@psychologie.uni-mannheim.de](mailto:kieslich@psychologie.uni-mannheim.de)  
[dirk.wulff@gmail.com](mailto:dirk.wulff@gmail.com)

Mousetrap-os plugins: <https://github.com/pascalkieslich/mousetrap-os>

Mousetrap R package: <http://pascalkieslich.github.io/mousetrap/>

Thanks:

Felix Henninger, co-developer of mousetrap-os plugin and R package

Jonas Haslbeck & Michael Schulte-Mecklenbeck, co-developers of mousetrap R package

Mila Rüdiger and Monika Wiegmann for data collection and testing