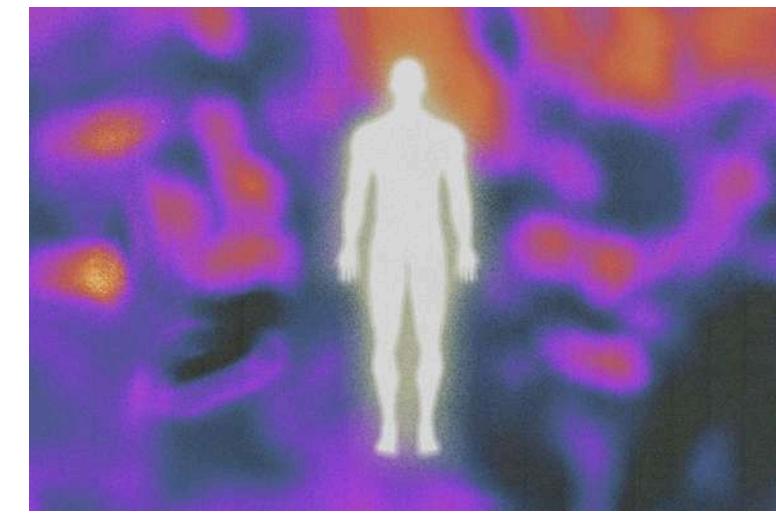
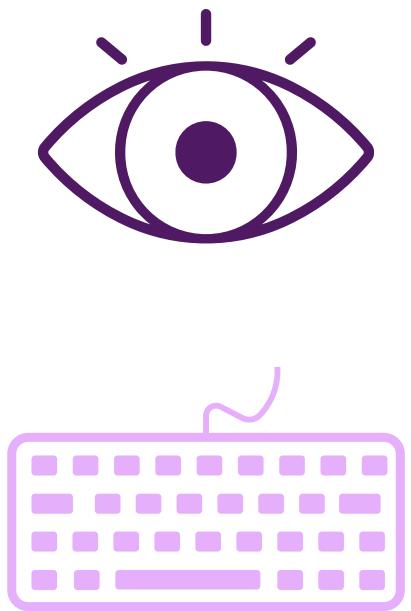
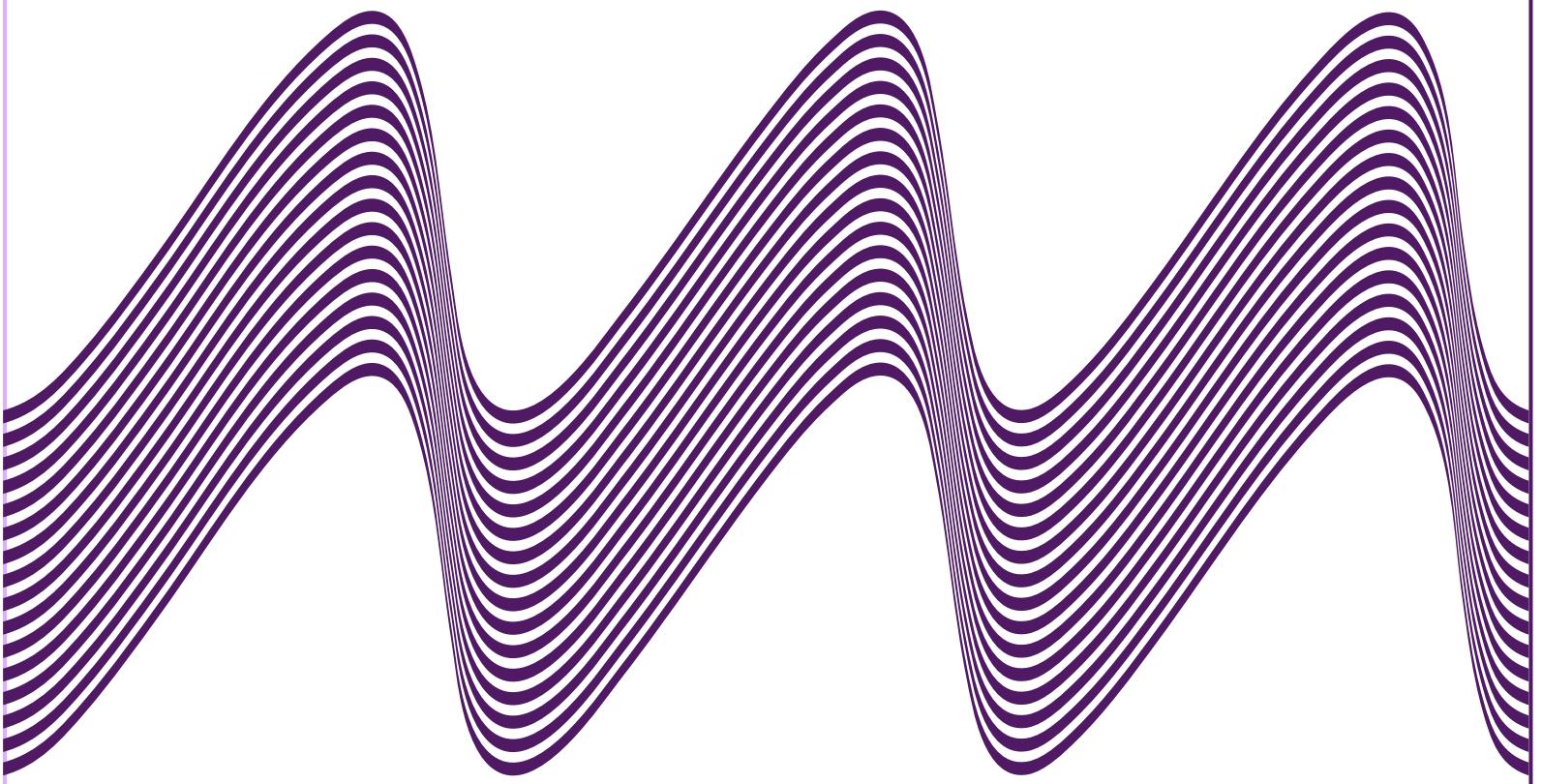




PSYCHOPY

PSYCHOLOGICKÉ EXPERIMENTY
SNADNO A RYCHLE





[Back to Agenda Page](#)

ZÁKLADY PC

Kde jsou krizové body? Latence,
refresh rate, color-accuracy ...



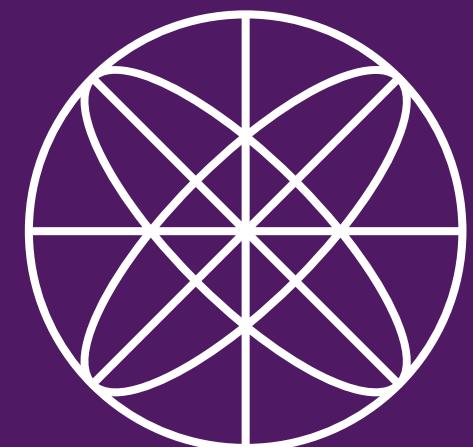
PSYCHOPY

Jak na experimenty?



PYTHON

minimální porozumění
programování



JASP

Jak na analýzu?

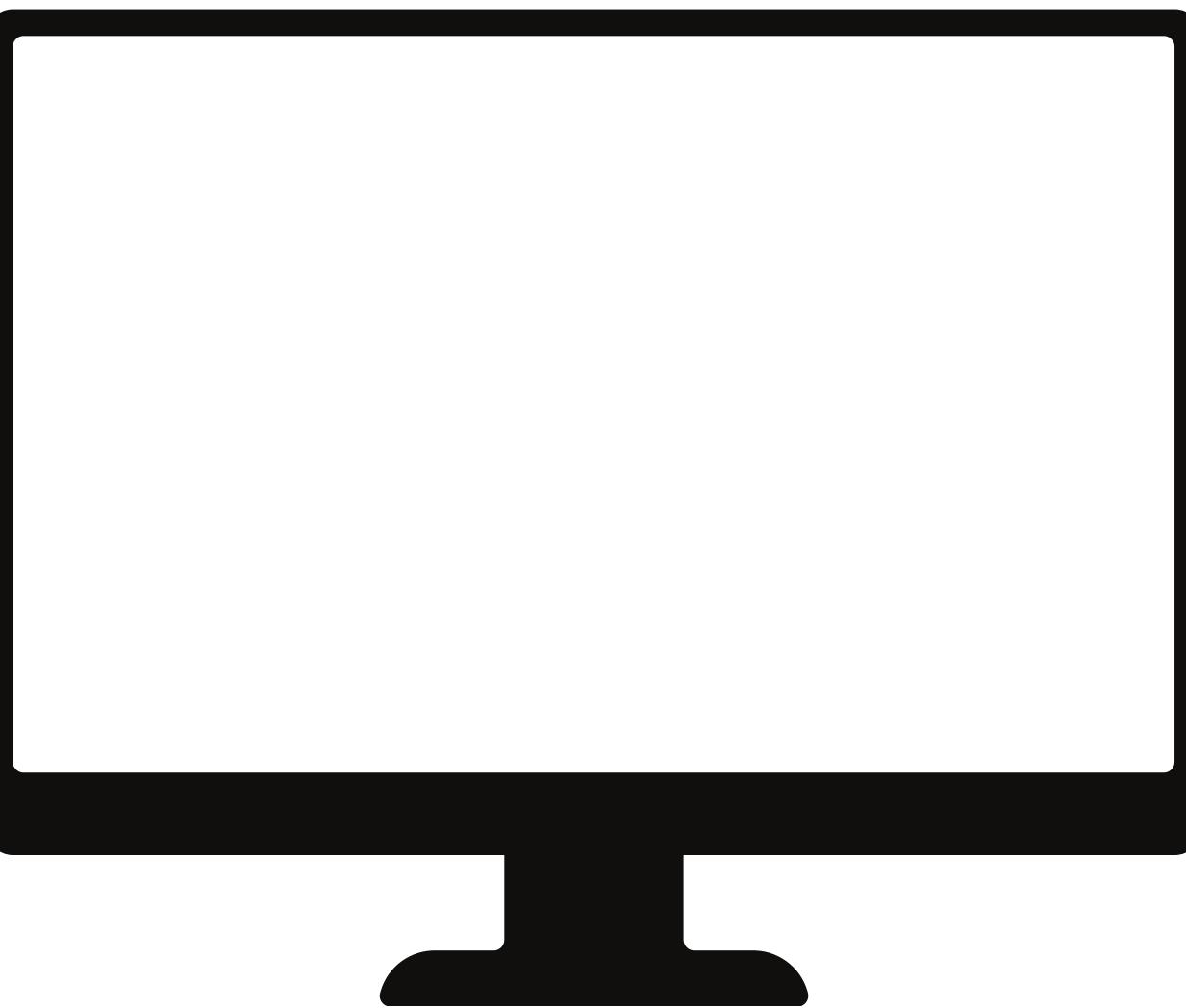
OTÁZKA

Presentovali jsme obličej na 100 ms

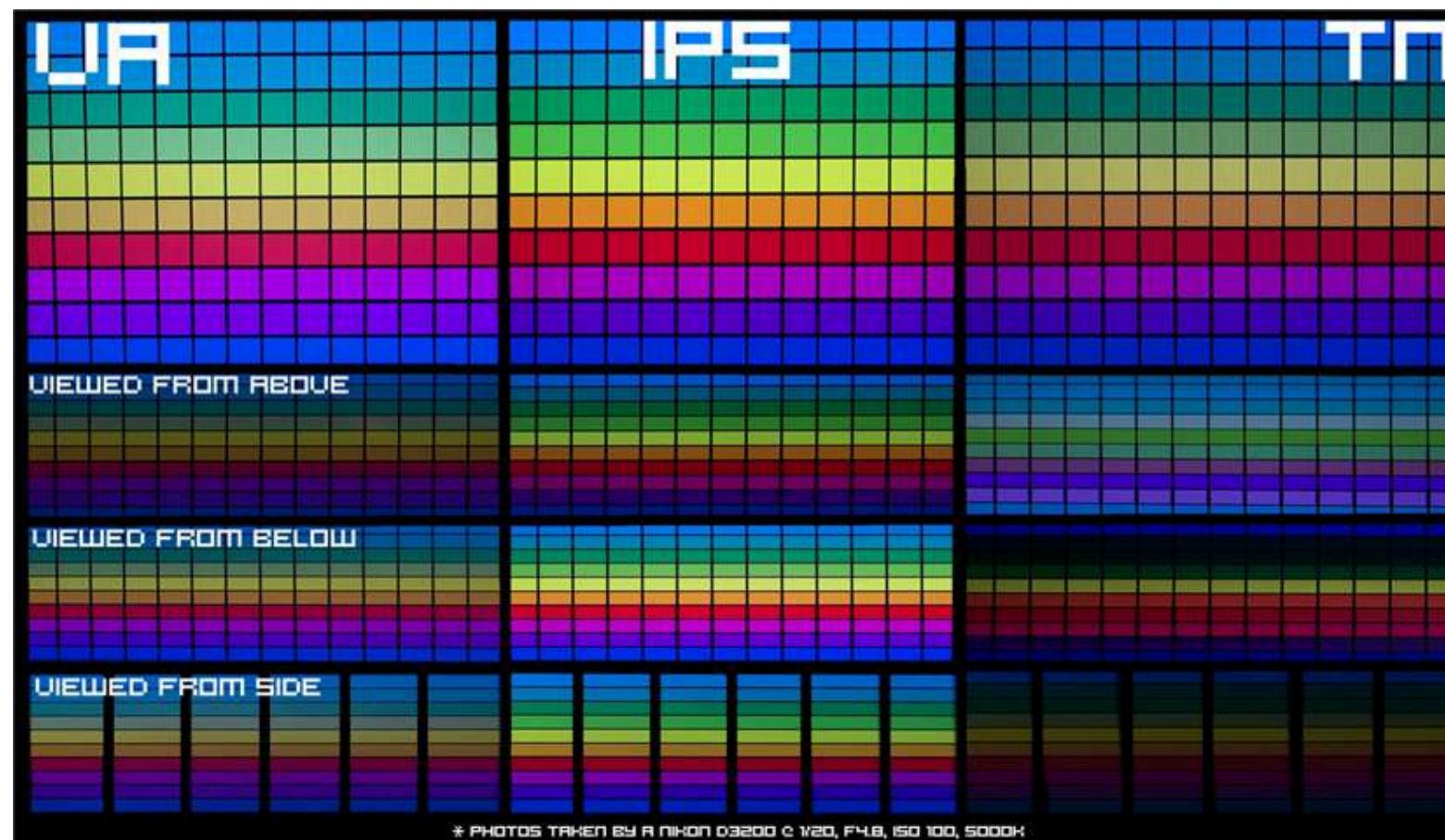
Participant má za úkol vybrat ze dvou obličejů

Měříme reakční čas a správnost odpovědi

MONITORY



MONITORY



Panel Type



TN



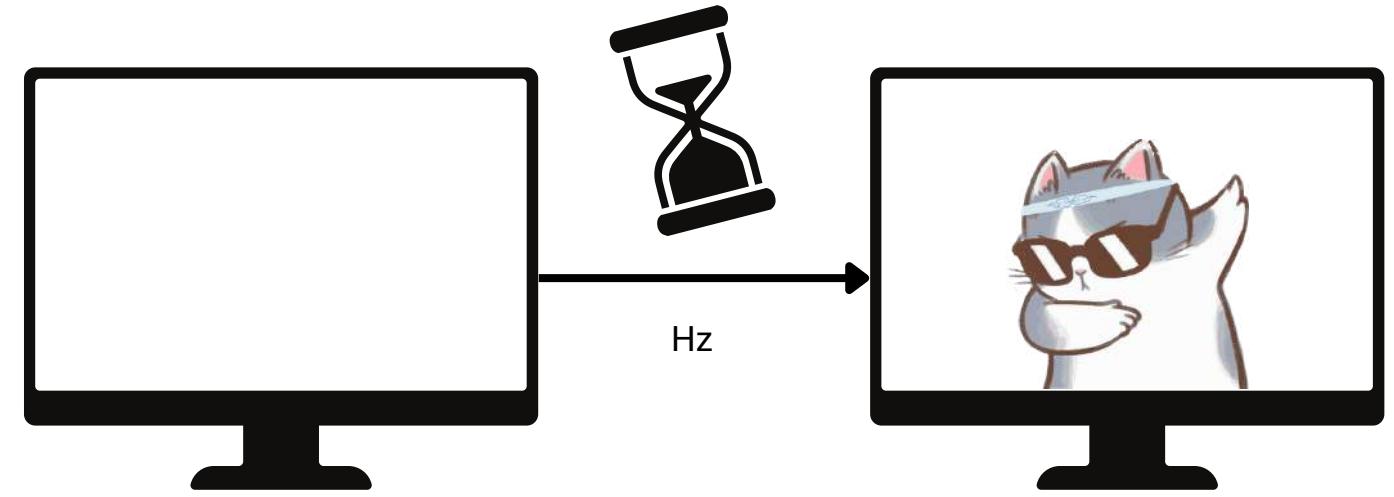
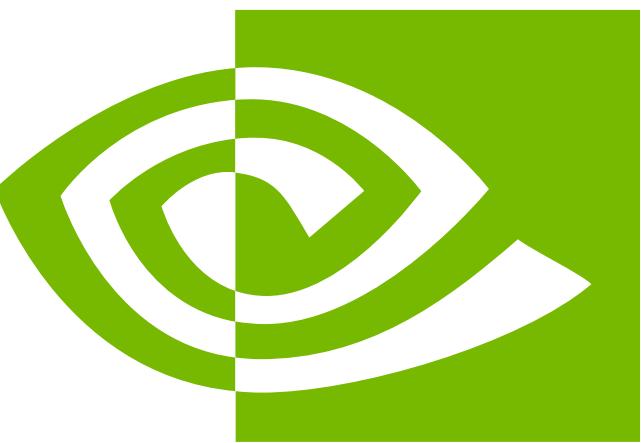
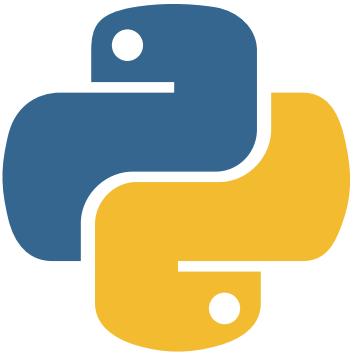
VA



IPS

Response Time	Often under 1ms	Often between 2-3ms	Often between 1-2ms
Colour	Limited colour space	Good	Best
Contrast	Good 1000:1	Best 3000:1	Good 1000:1
Viewing Angle	170/160	178/178	178/178

RENDER PATH



Prosím tohle



REFRESH RATE

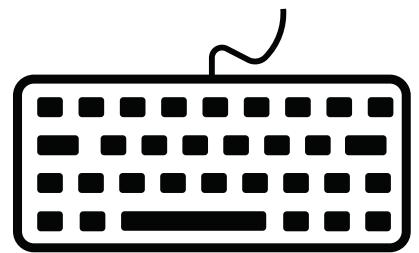
Jak rychle se může váš
displej změnit



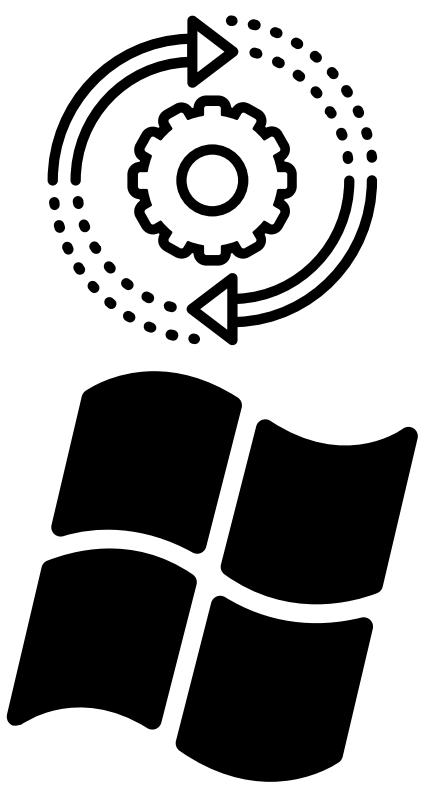
Blur Busters TestUFO Motion Tests. Benchmark for monitors & displays.

Blur Busters UFO Motion Tests with ghosting test, 30fps vs 60fps vs 120hz vs 144hz vs 240hz, PWM test, motion blur test, judder test, benchmarks, and more.

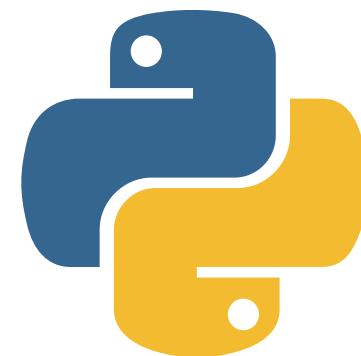
LATENCE KLÁVESNICE



refresh (polling) rate



repetice
double pressess

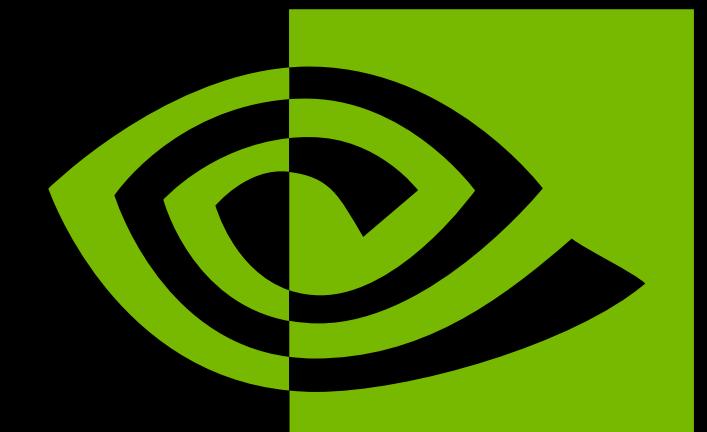
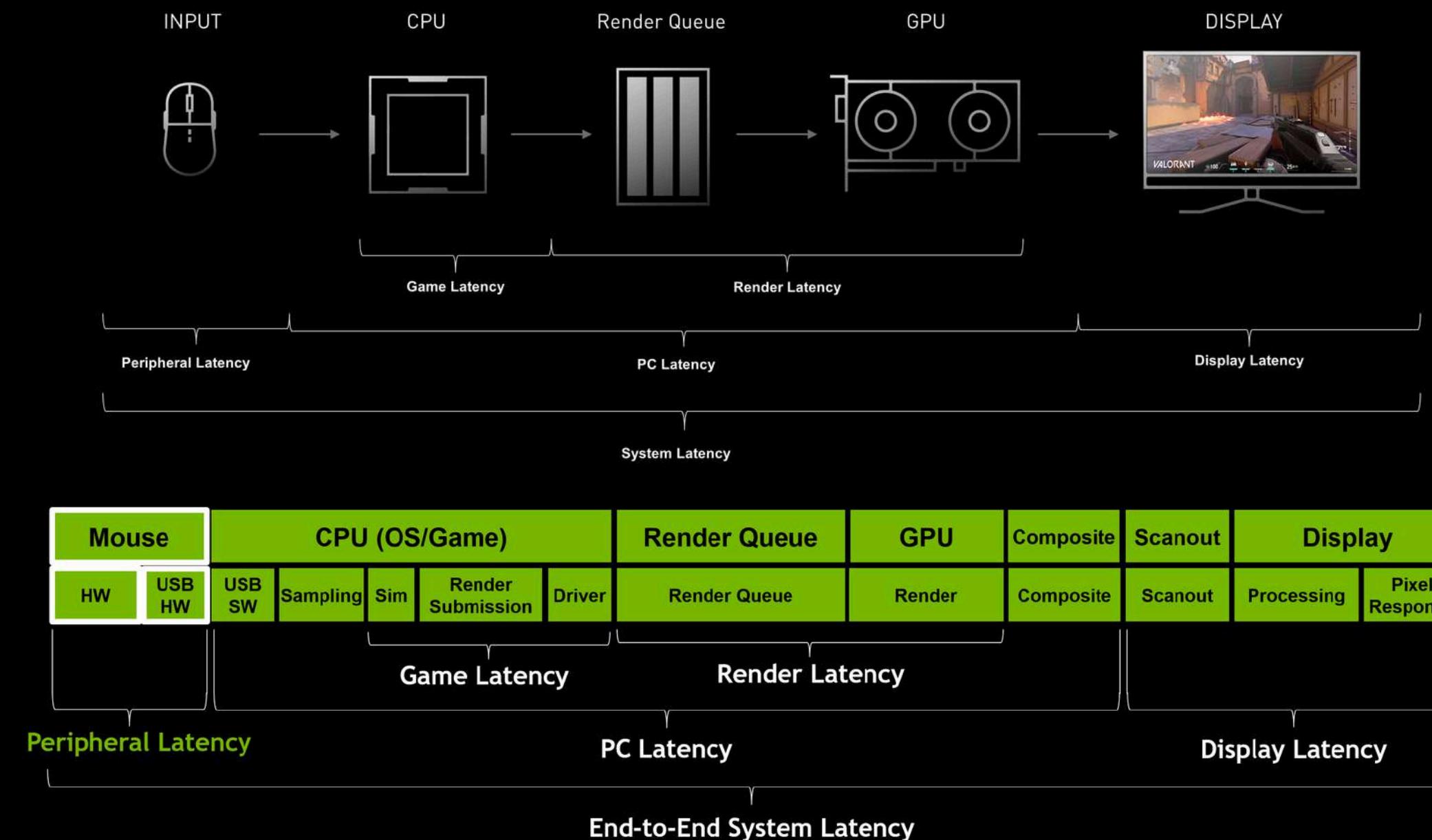


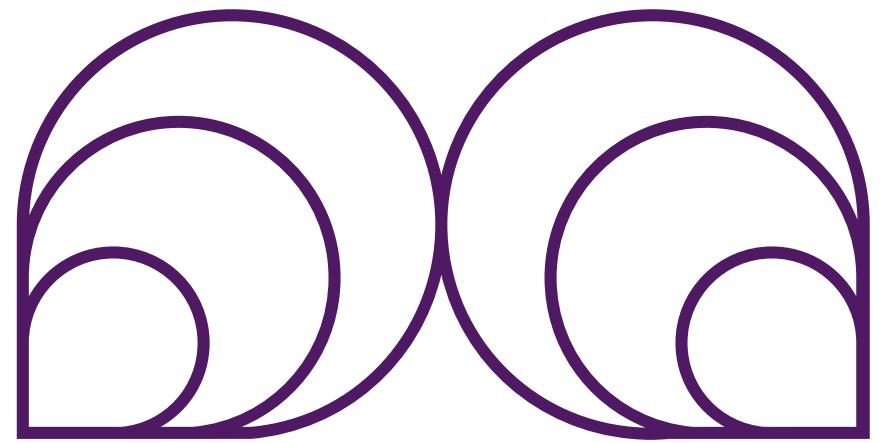
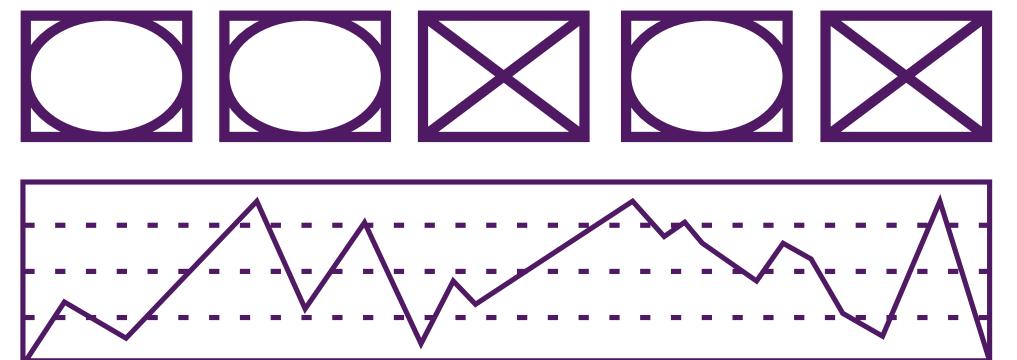
wireless - no, thank you 😊

LATENCE

SYSTEM LATENCY TERMINOLOGY

<https://www.nvidia.com/en-us/geforce/guides/gfecnt/202010/system-latency-optimization-guide/>





PSYCHOPY

CO TO JE?

EXPERIMENTÁLNÍ BALÍČKY

Psychtoolbox

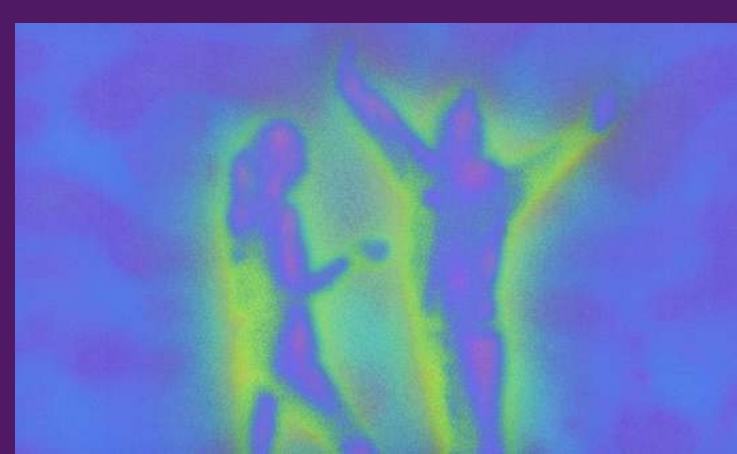
JsPsych

OpenSesame

E-prime

PsychoPy

CO PSYCHOLOGY UMÍ?



AUTOMATICKÉ LOGY

Nemusíme kódovat vlastní řešení

NÁSTROJE NA ŘEŠENÍ TYPICKÝCH PROBLÉMŮ

Latence, kalibrace monitoru

REPRODUCIBILITU

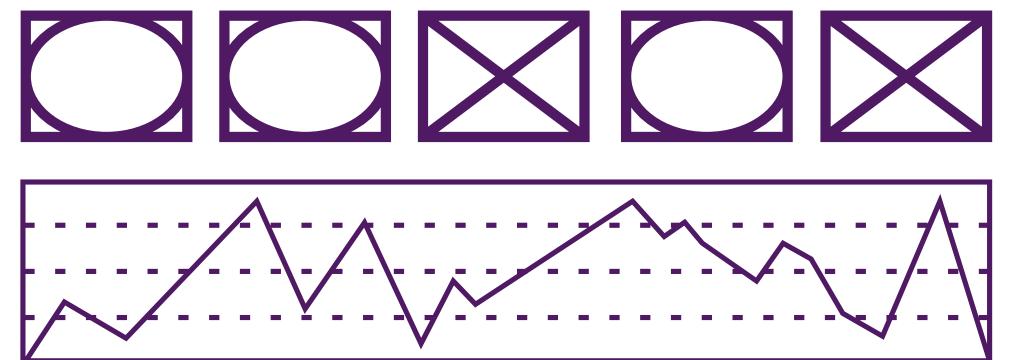
Je snadné distribuovat experimenty a administrovat po celém světě

EXPERIMENTY DEMO

Demo experimenty (uvnitř psychopy)

Pavlovia (psychopy online)

OSF (např. <https://osf.io/6b7ke/>)



600



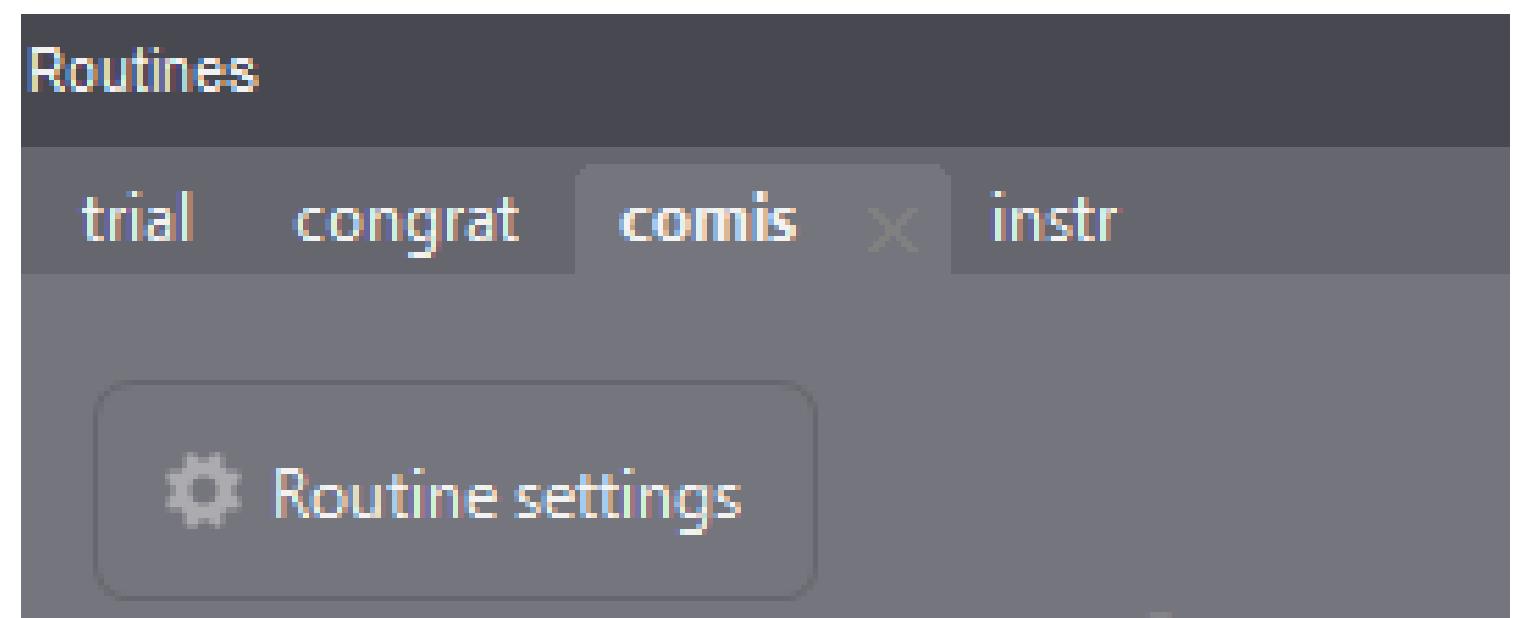
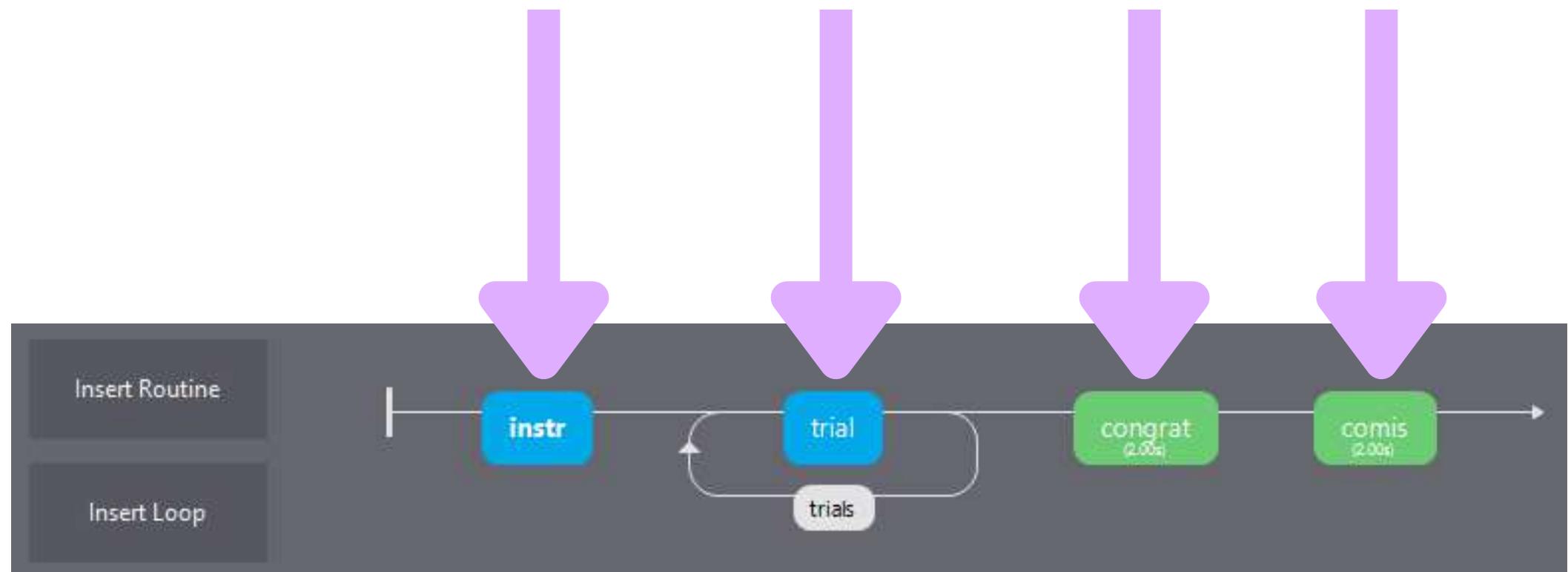
PSYCHO PY

ZÁKLADNÍ STAVEBNÍ KAMENY

ROUTINE

Lze si představit jako jednu “obrazovku”

Složená z komponent (obraz, text, input atd.)



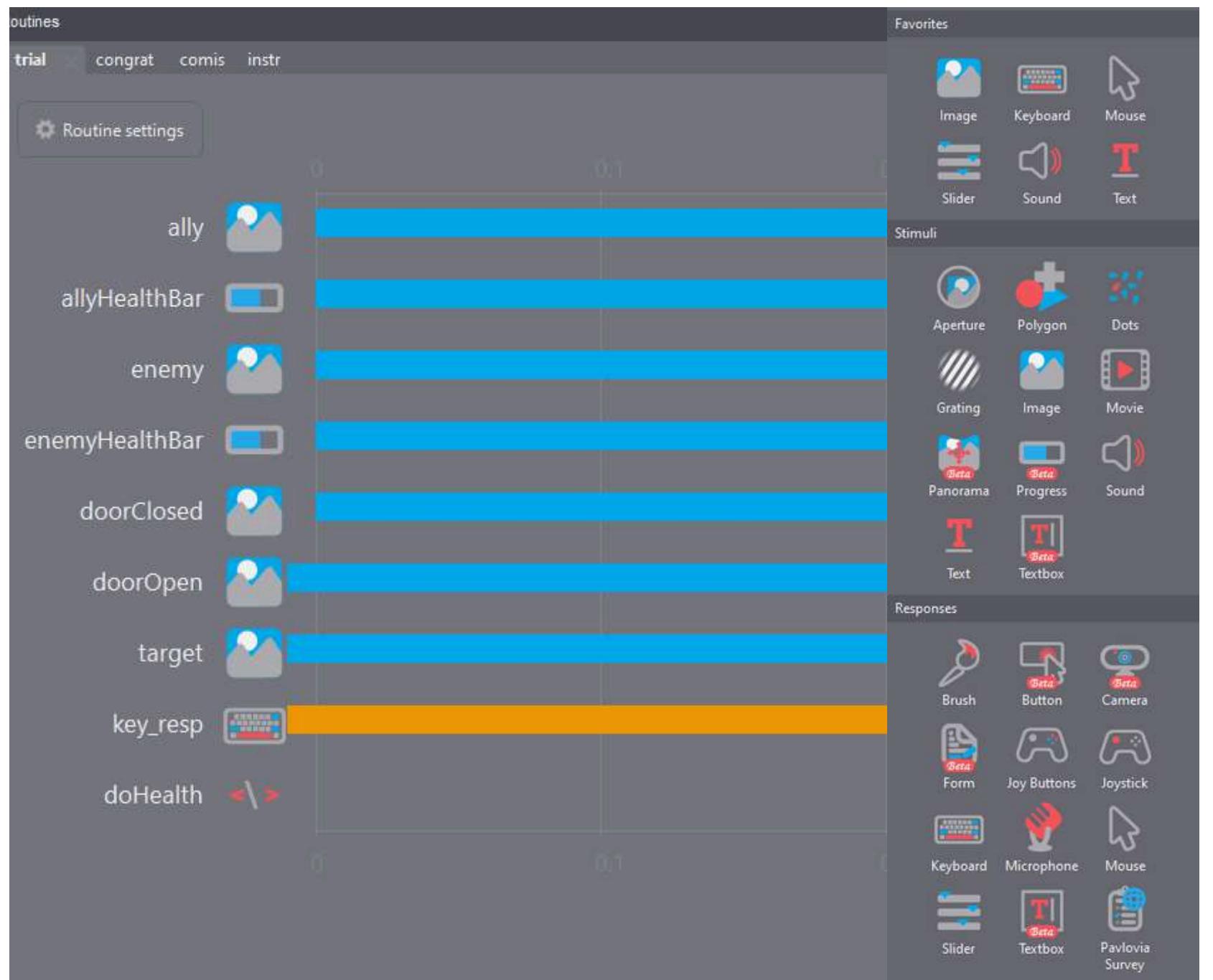
COMPONENT

Samotný obsah experimentu

Python kód umožňující

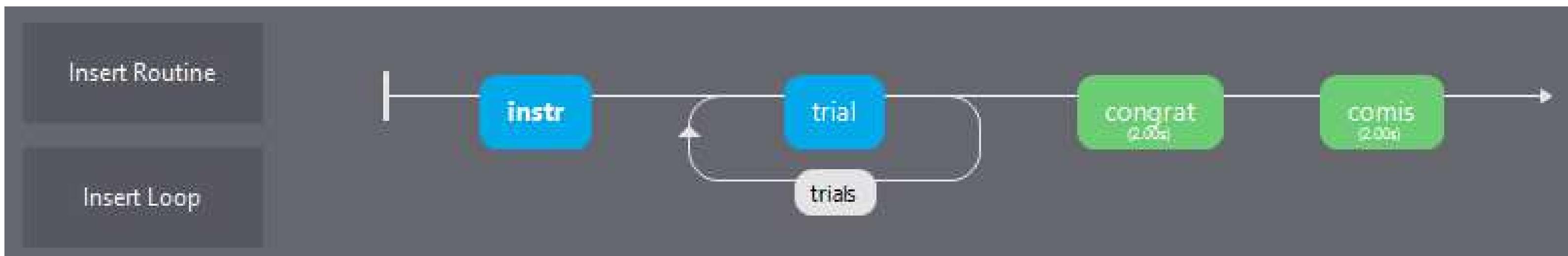
- zobrazit text
- zobrazit obrázek
- poslouchat klávesy

Pořadí komponent ukazuje pořadí ve kterém se "kreslí"



EXPERIMENT

Kombinace všech Routines + nastavení atd.



TEXT COMPONENT

Zobrazování textových stimulů,
informovaného souhlasu, instrukcí,
feedbacku a poděkování.

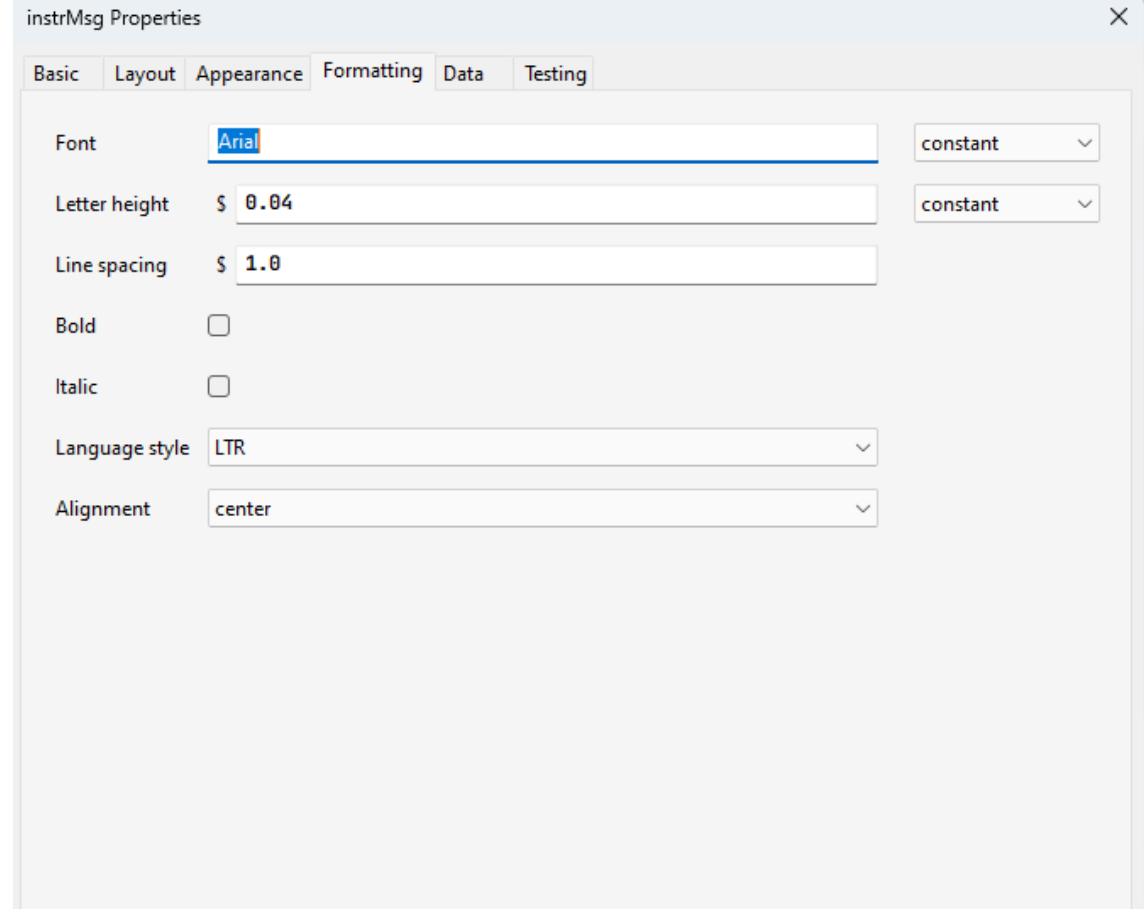
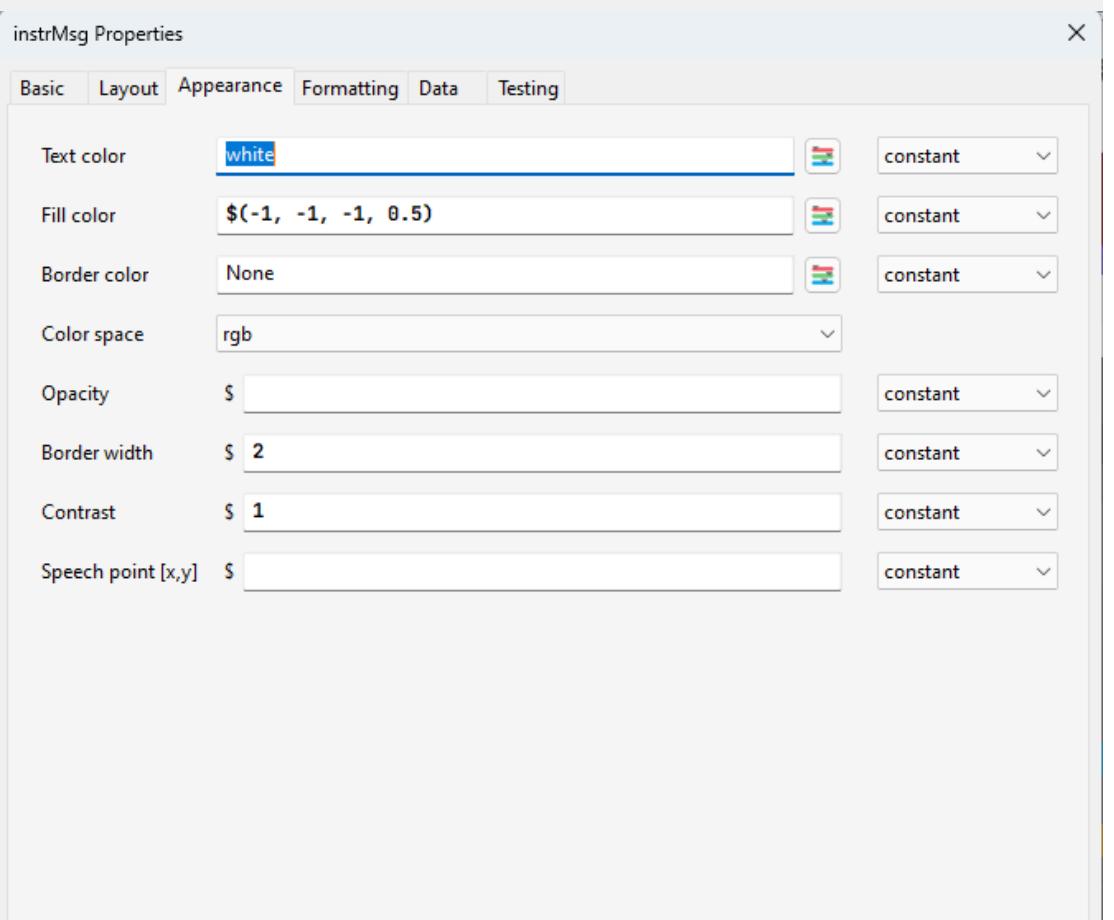
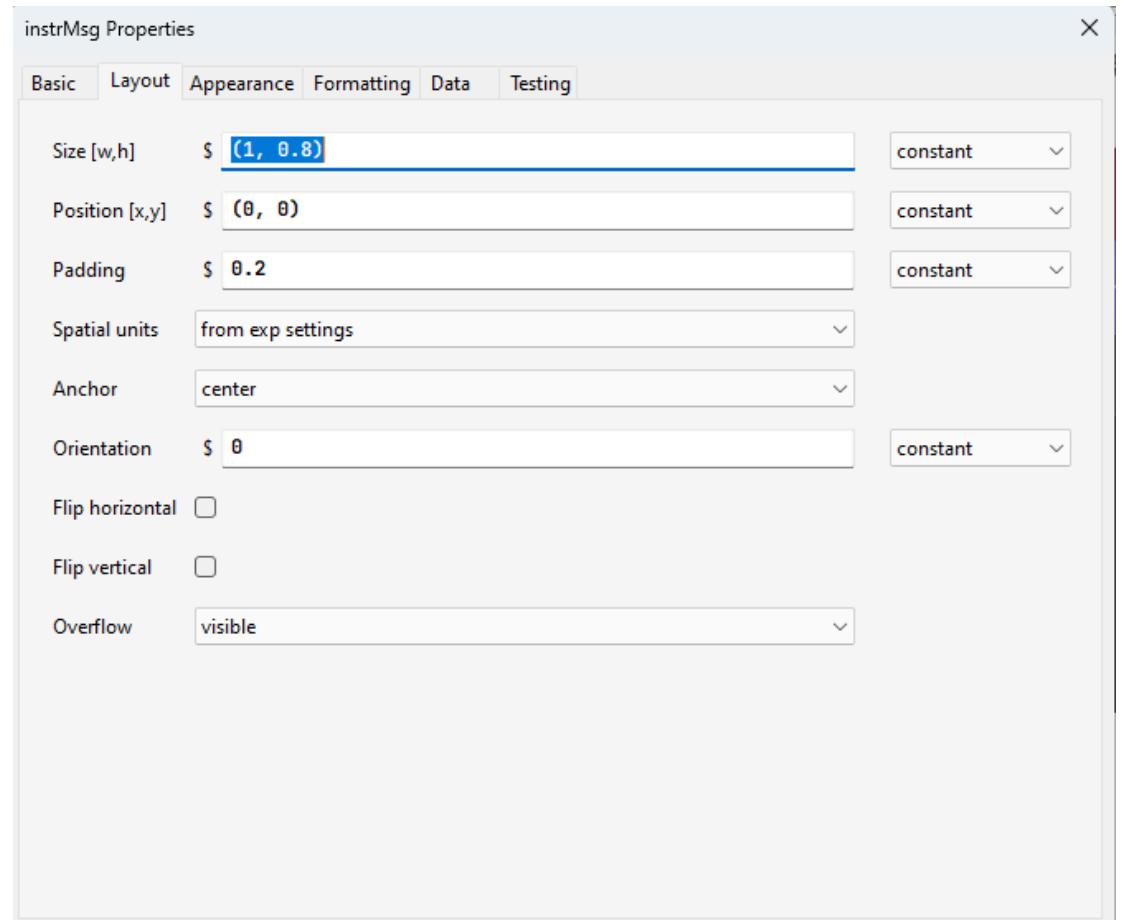
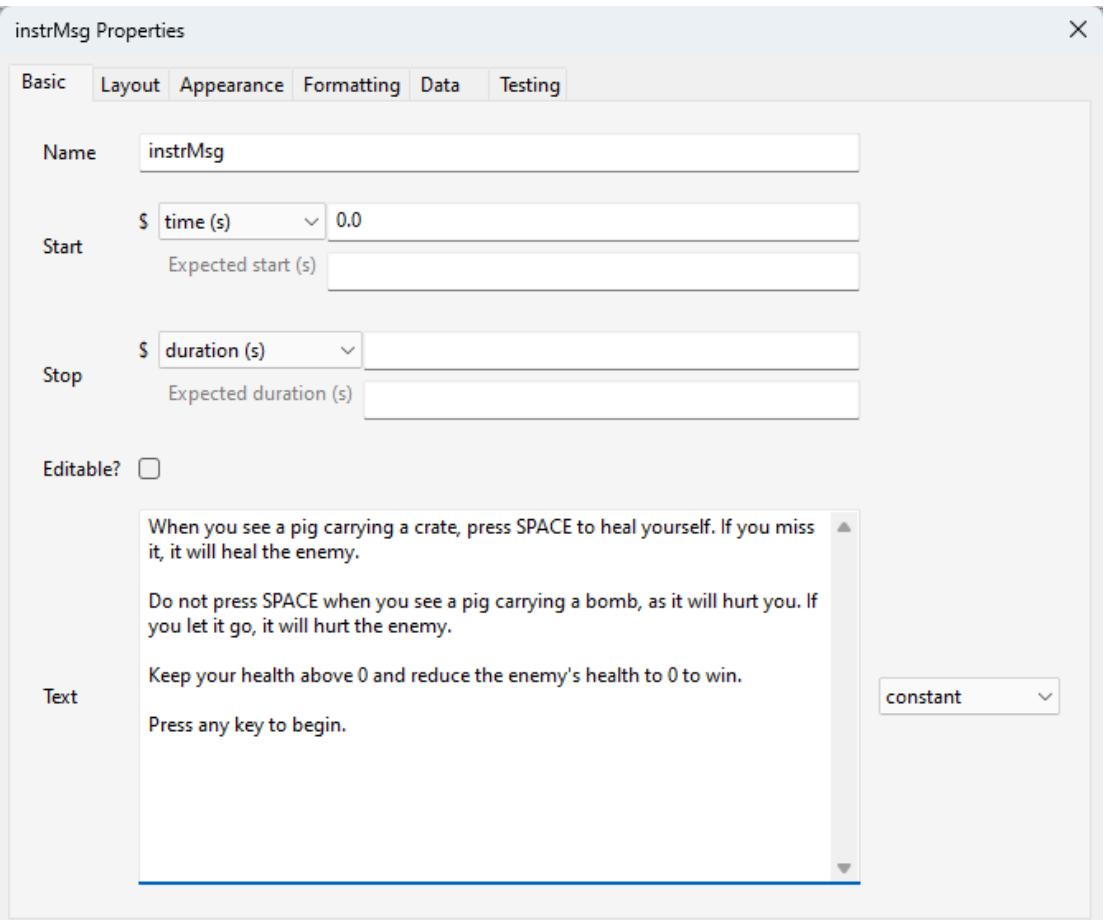
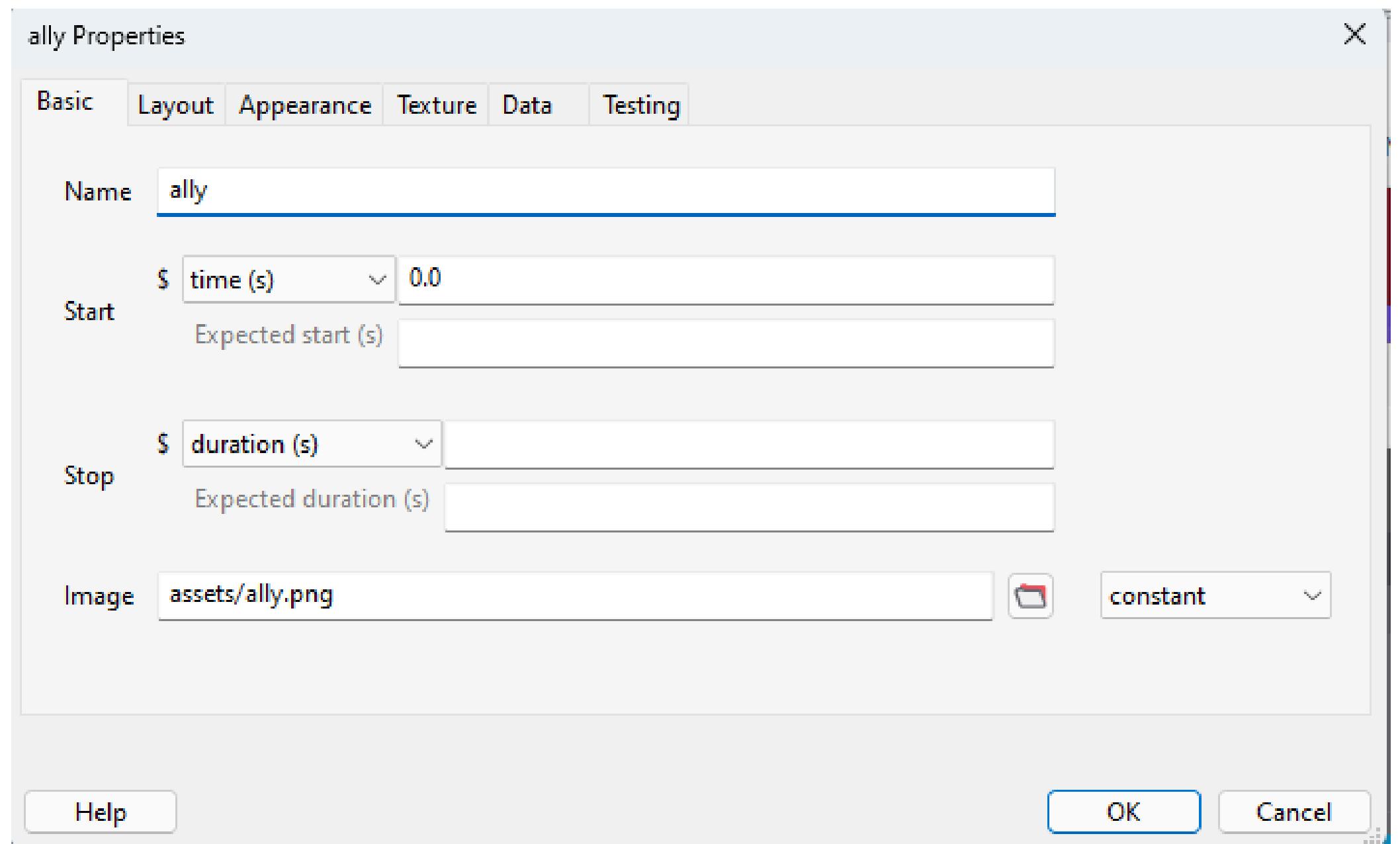


IMAGE COMPONENT

Zobrazování obrazových stimulů



KEYBOARD COMPONENT

Sběr odpovědí pomocí klávesnice

key_resp Properties

Basic Data Testing

Name: key_resp

Start: \$ time (s) 2
Expected start (s):

Stop: \$ duration (s)
Expected duration (s):

Force end of Routine:

Register keypress on...: press

Allowed keys: \$ 'space'
constant

key_resp Properties

Help Basic Data Testing

Store: last key

Store correct:

Correct answer:

Save onset/offset times: What is the 'correct' key? Might be helpful to add a correctAns column and use \$correctAns to compare to the key press.

Sync timing with screen:

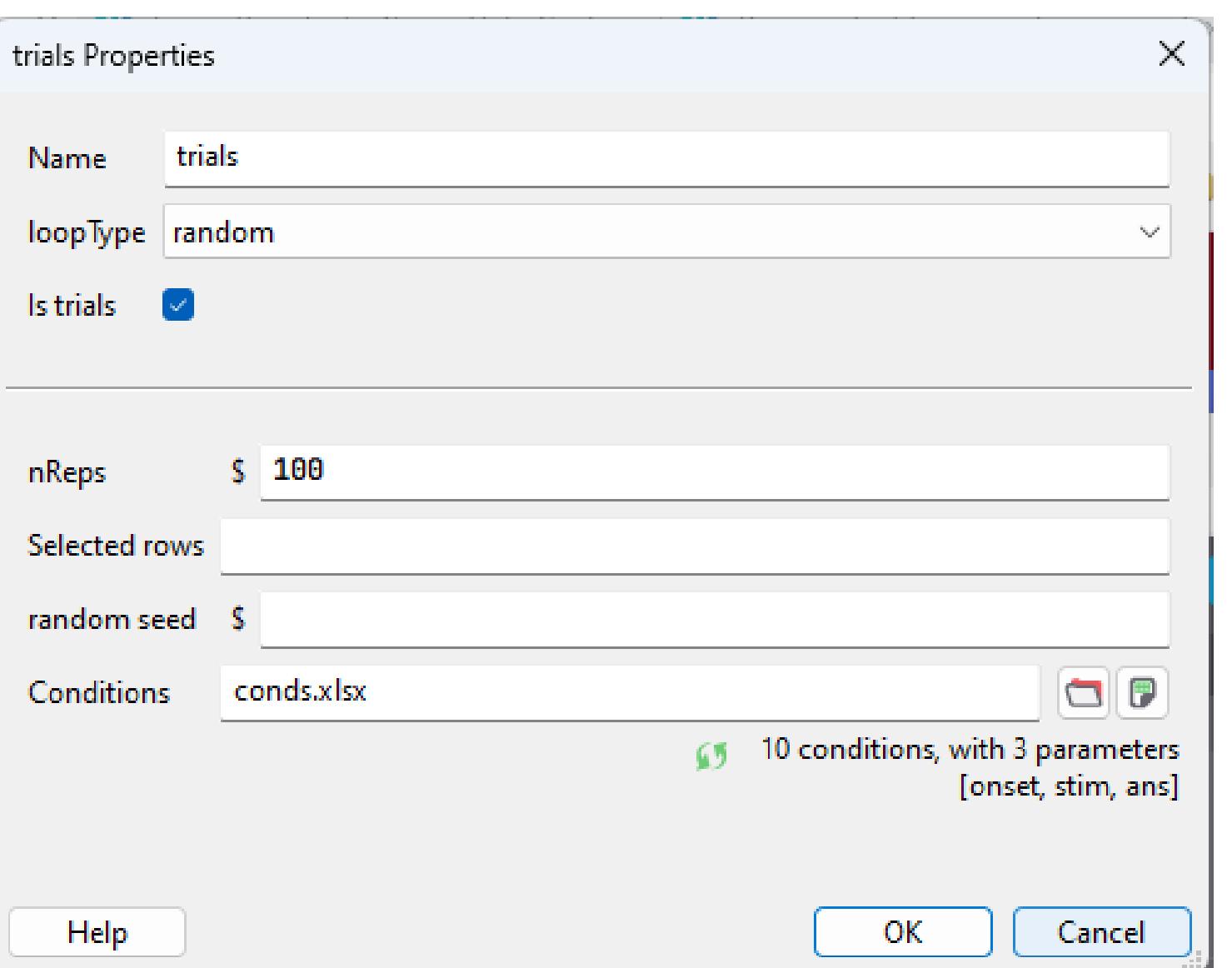
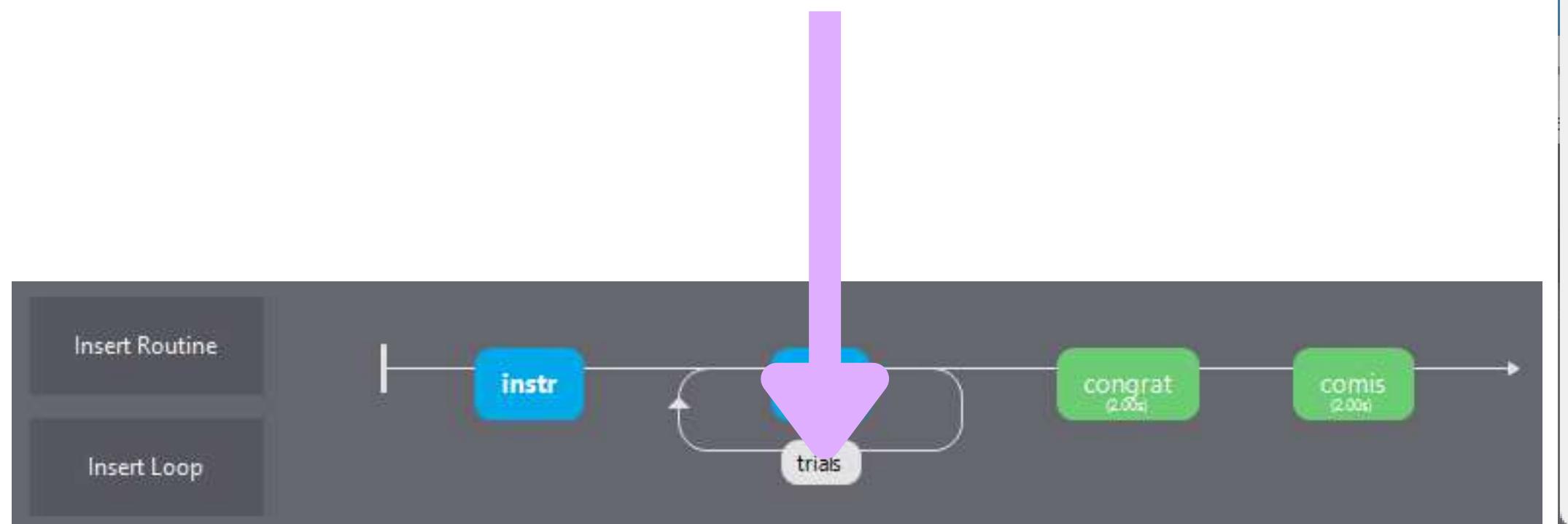
Discard previous:

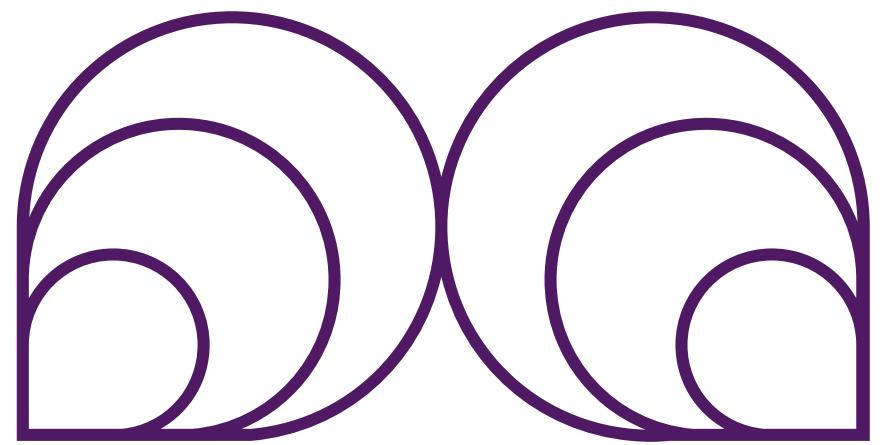
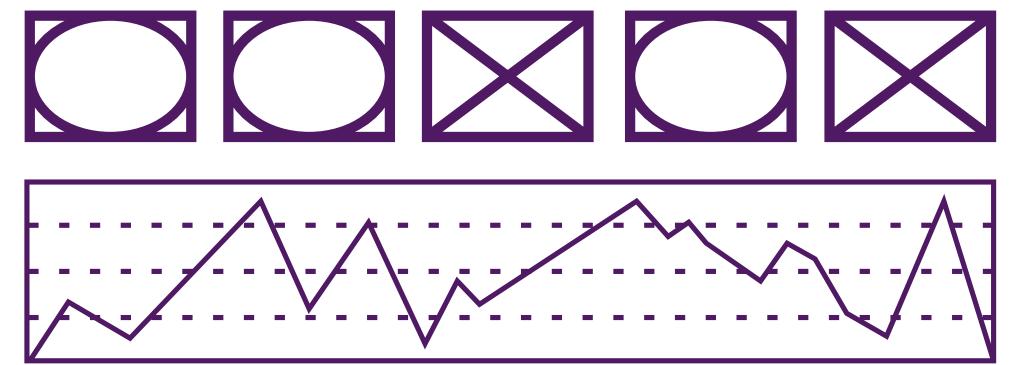
Help OK Cancel

LOOP

Opakování konkrétní Routine

Základní stavební blok experimentu





PSYCHO PY

PRVNÍ

EXPERIMENT

PROCES

1. Design jednoho trialu
2. Experiment s jedním opakováním
3. Co se bude měnit?
4. Příprava ukázkového nastavení
5. Příprava plného nastavení

MĚŘÍME REAKČNÍ ČAS



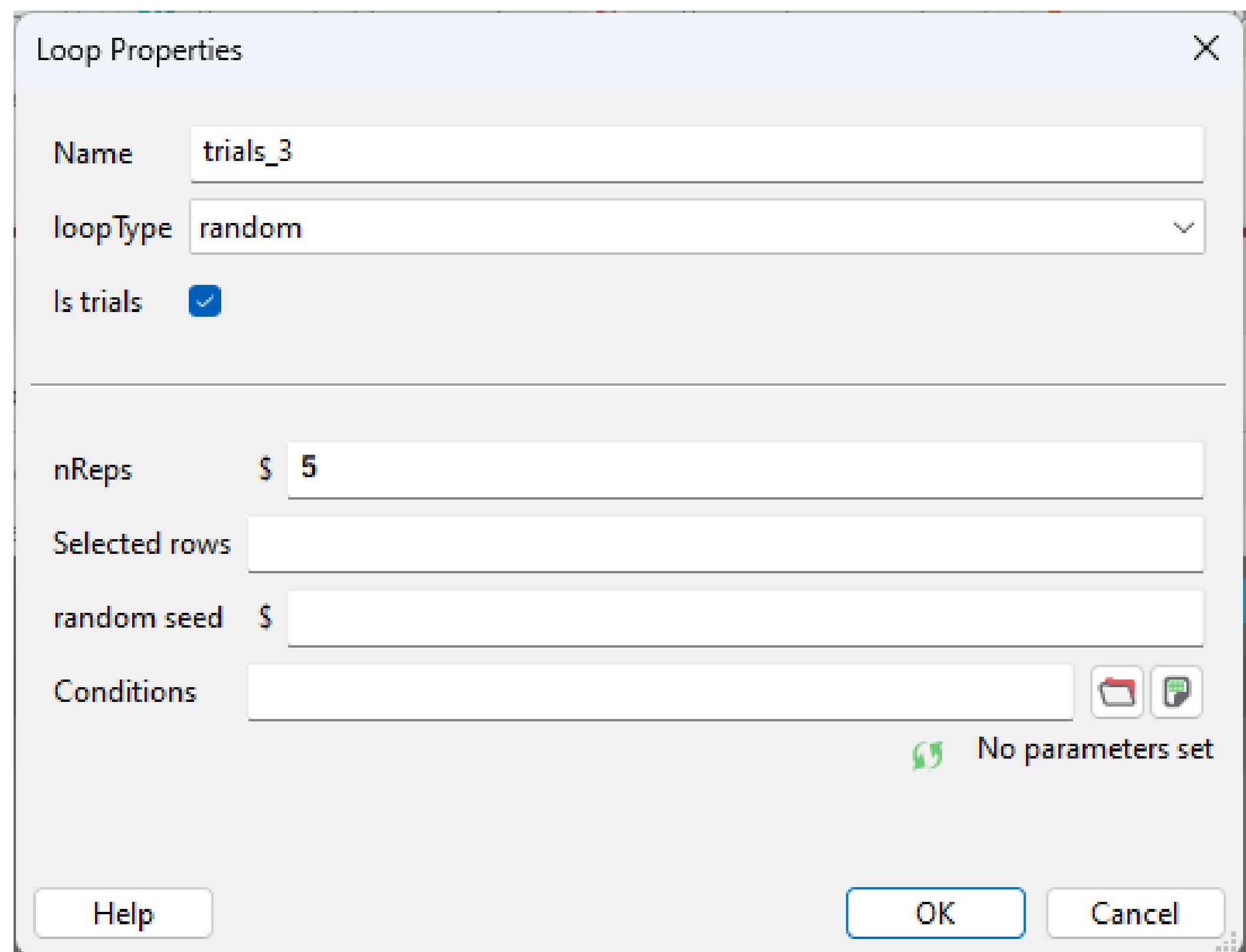
Design prvního trialu

1. Participant se dívá na červenou tečku.
2. Po 6ti s se změní na zelenou.
3. Stiskem mezerníku se zaregistruje odpověď



ROUTINE REPETITION

Opakujeme trial 5krát

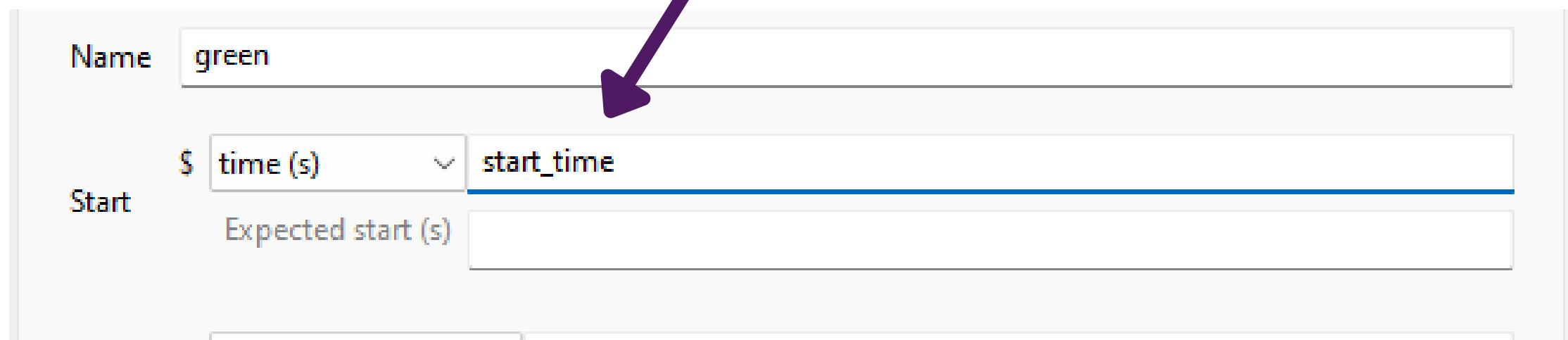
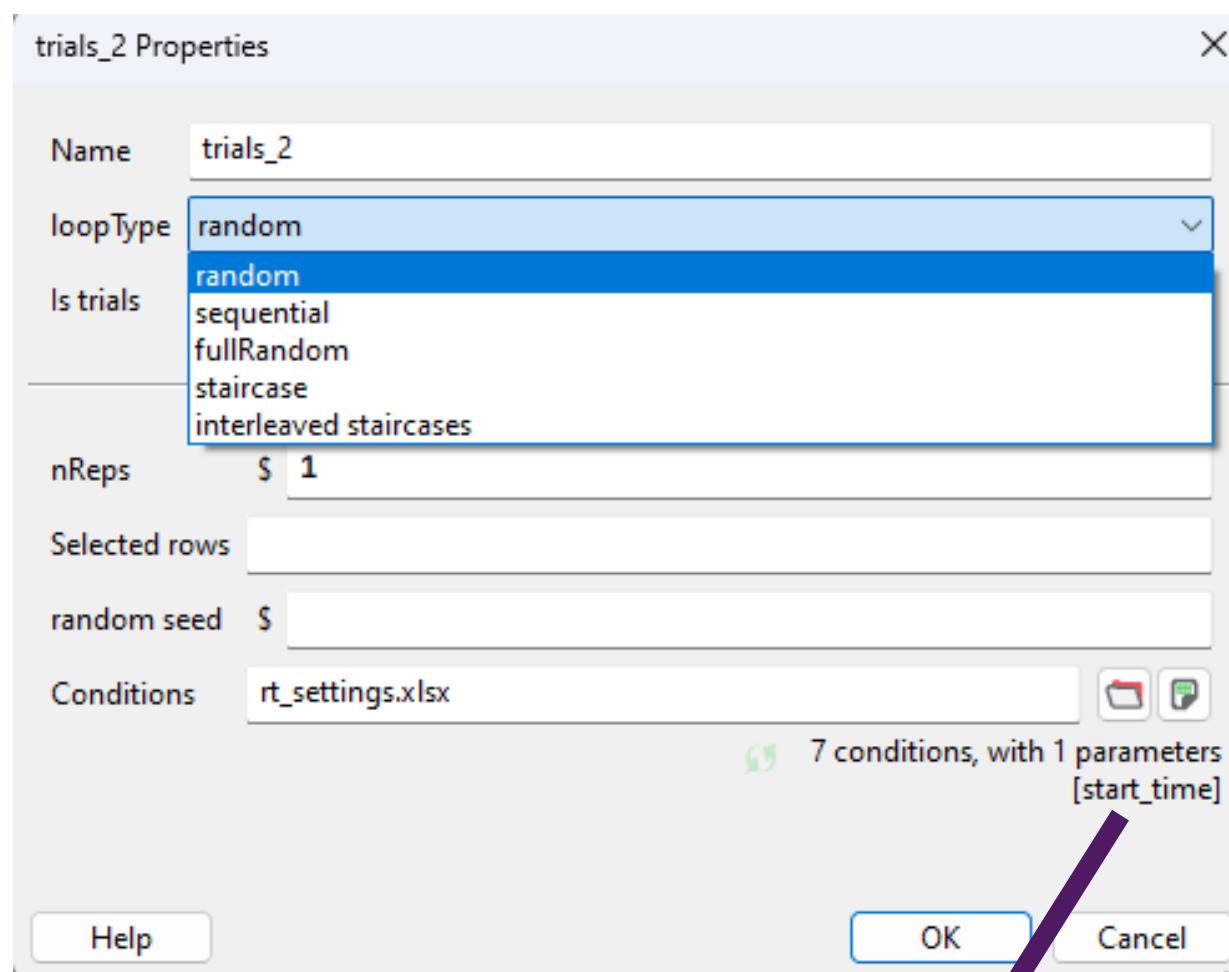


ROUTINE SETTINGS

Jak změnit trial od jednoho k druhému

Cokoli má v názvu \$ je proměnná (python funkce)

Nastavení jednotlivých trialů v Routine můžeme definovat souborem s nastavením



MĚŘÍME REAKČNÍ ČAS



Design vícero trialů

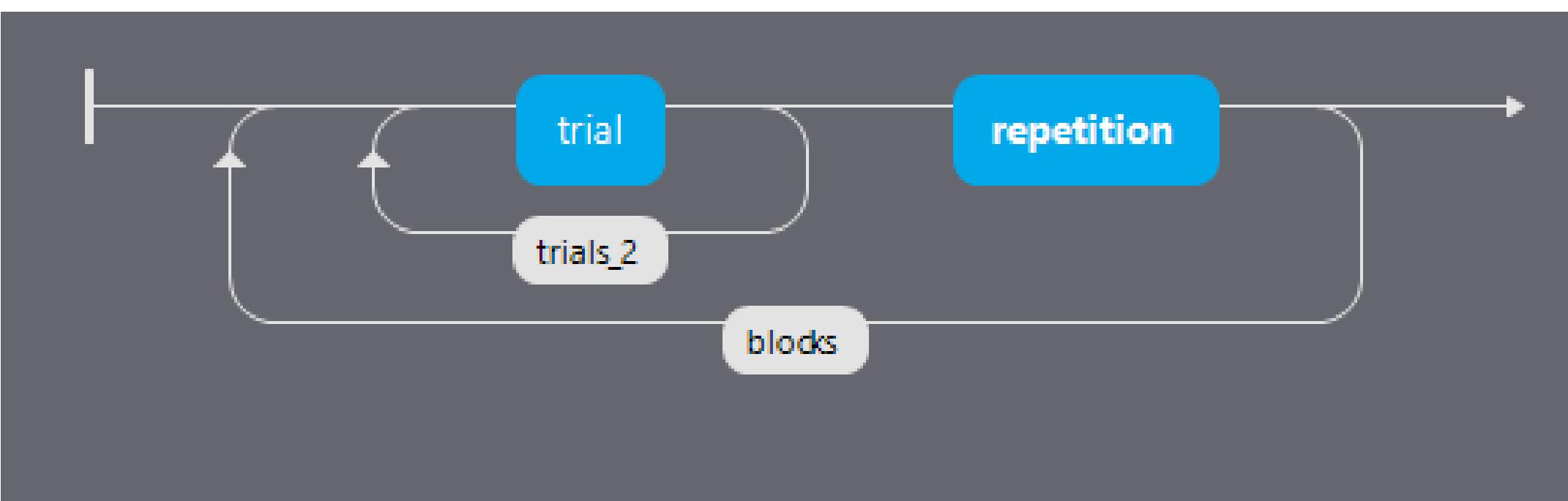
1. Participant se dívá na červenou tečku.
2. Po X s se změní na zelenou
 - a. X je mezi 1 a 5 ti s.
3. Stiskem mezerníku se zaregistruje odpověď.
4. Celkem 25 trialů.
5. Přidejte úvodní slide a děkovací slide



NESTING LOOPS

Umožňuje nám presentovat stimuly ve více separátních blocích s pauzou

Bloky mohou mít vlastní nastavení a vlastní proměnné v settings (např. time multiplier, difficulty etc.)

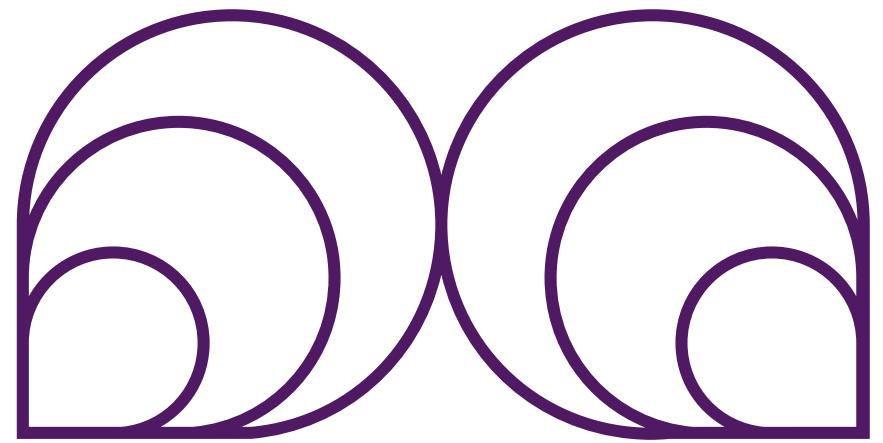
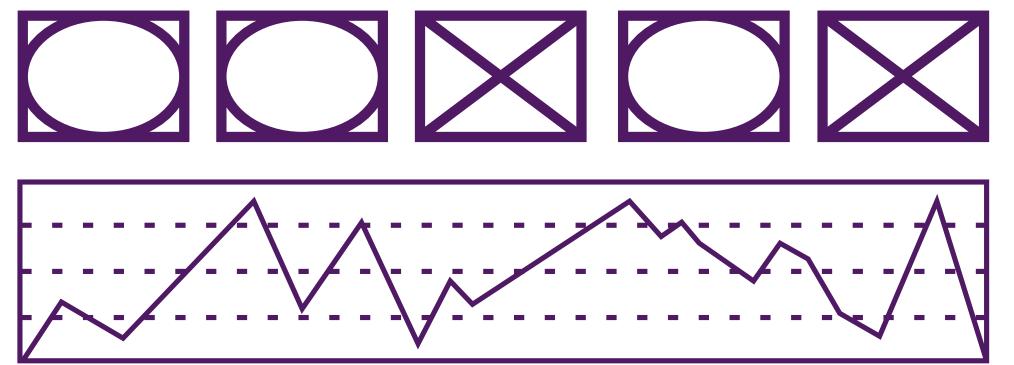


MĚŘÍME REAKČNÍ ČAS



1. Po 5 ti trialech pauza s informací, že experiment bude ještě pokračovat





PSYCHOPY

DATA Z

PSYCHOPY

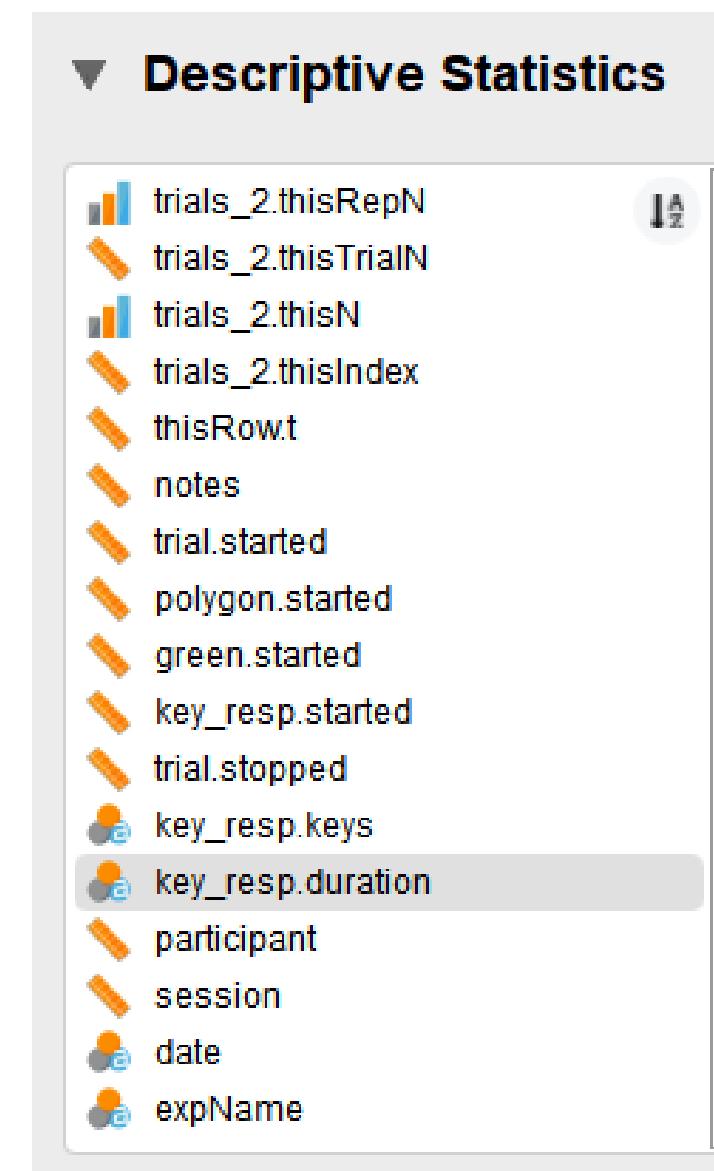
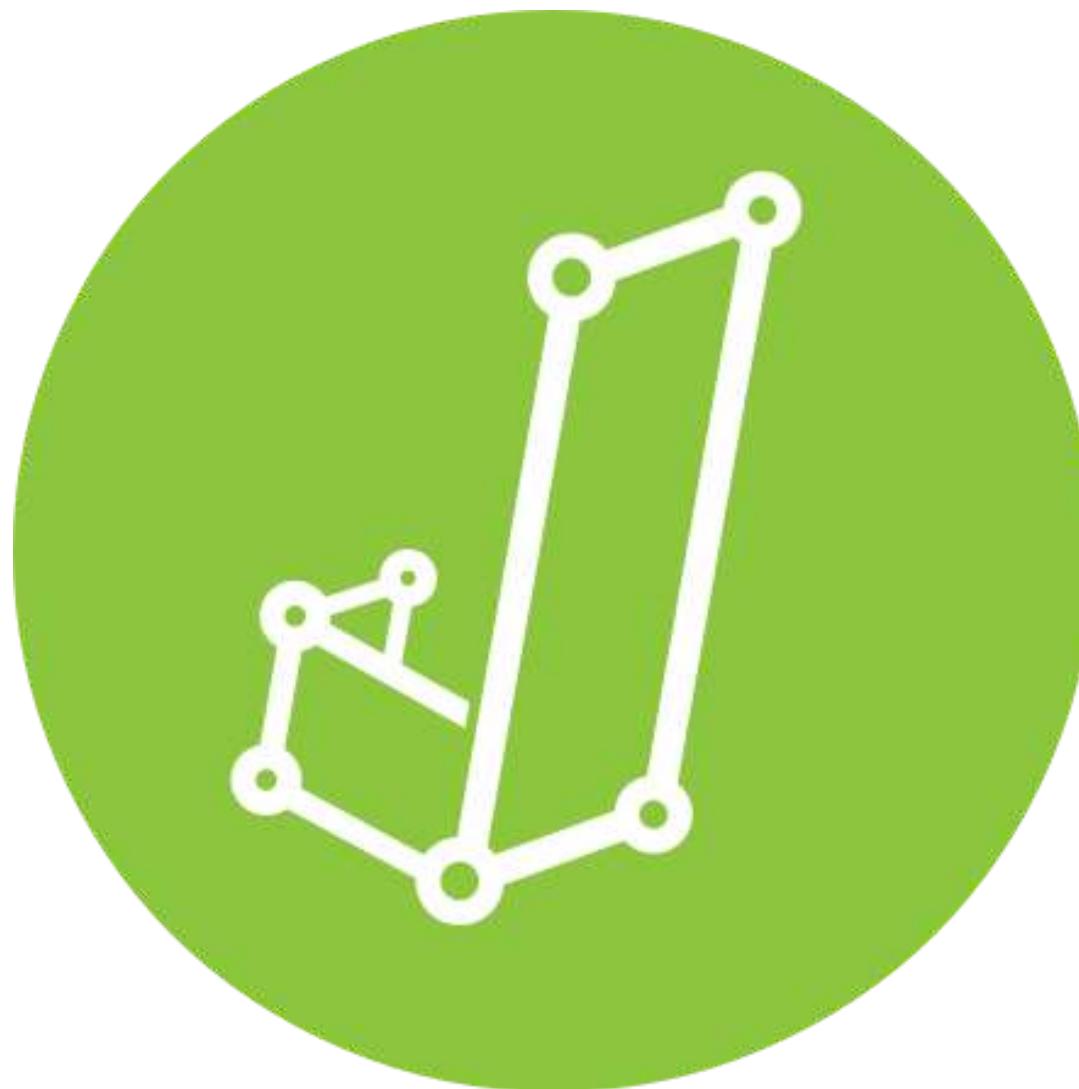
DATOVÉ VÝSTUPY

Je nutné nejprve spojit data dohromady

Výstupy jsou CSV soubory, lze udělat online, v excelu či jiných nástrojích

 400868_reaction_time_2023-11-29_13h58.09.958.csv	29.11.2023 13:58	Comma Separate...	2 KB
 400868_reaction_time_2023-11-29_13h58.09.958.log	29.11.2023 13:58	Text Document	4 KB
 400868_reaction_time_2023-11-29_13h58.09.958.psydat	29.11.2023 13:58	PSYDAT File	4 KB
 452143_reaction_time_2023-11-29_13h55.44.536.csv	29.11.2023 13:55	Comma Separate...	1 KB

ANALÝZA

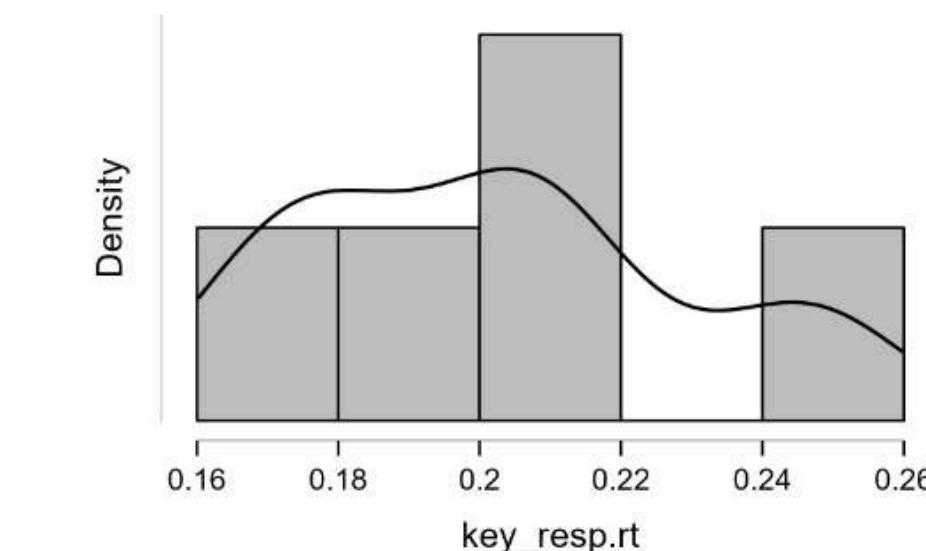


Descriptive Statistics

Descriptive Statistics	
	key_resp.rt
Valid	5
Missing	0
Mean	0.202
Std. Deviation	0.029
Minimum	0.170
Maximum	0.246

Distribution Plots

key_resp.rt



VÍCE PARTICIPANTŮ

Je nutné nejprve spojit data dohromady

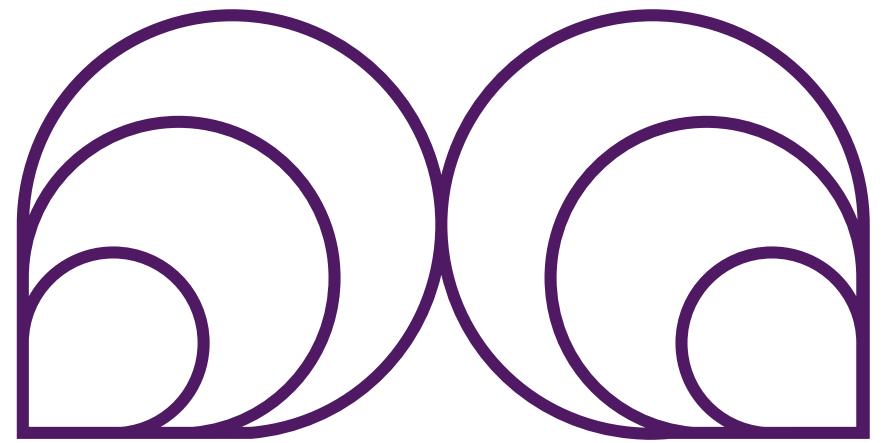
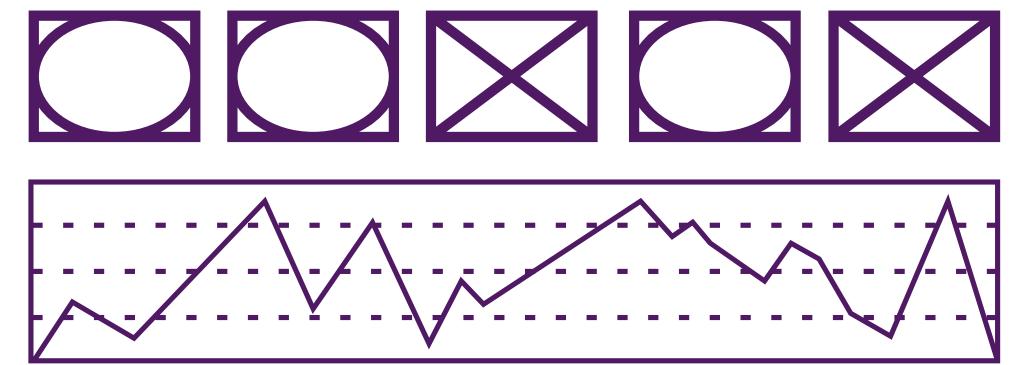
Výstupy jsou CSV soubory, lze udělat online, v excelu či jiných nástrojích

Merge CSV files online into one file

Use this tool to merge multiple csv files into one. This combined file can be imported to Excel etc.

merge-csv.com

participant	session	date	expName	psychopyVersion	frameRate	expStart	v
400868	1	2023-11-29_13h58.09.958	reaction_time	2023.2.3	143.8401877	2023-11-29 13h58.14.512998 +0100	
400868	1	2023-11-29_13h58.09.958	reaction_time	2023.2.3	143.8401877	2023-11-29 13h58.14.512998 +0100	
400868	1	2023-11-29_13h58.09.958	reaction_time	2023.2.3	143.8401877	2023-11-29 13h58.14.512998 +0100	
400868	1	2023-11-29_13h58.09.958	reaction_time	2023.2.3	143.8401877	2023-11-29 13h58.14.512998 +0100	
400868	1	2023-11-29_13h58.09.958	reaction_time	2023.2.3	143.8401877	2023-11-29 13h58.14.512998 +0100	



PYTHON
CO TO JE?

KDE?

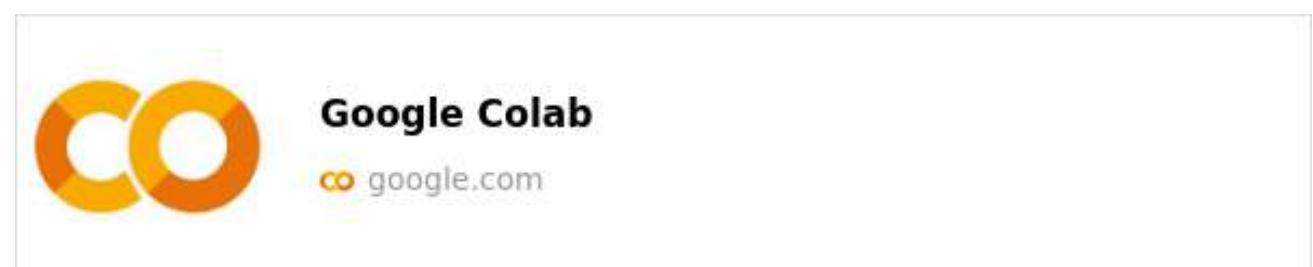
```
Shell | Output
Python shell in PsychoPy (pid:35372) - type some commands!
Python 3.8.10 (tags/v3.8.10:3d8993a, May 3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.

>>> |
```

A screenshot of a code editor interface. On the left, there's a sidebar with a search bar, a 'Files' section containing 'main.py', and 'Packager files' with 'poetry.lock' and 'pyproject.toml'. The main area shows a file named 'main.py' with the number '1' at the top. A 'Run' button is visible at the top right. Below the code area, there's a message: 'Results of your code will appear here when you Run the project.' At the bottom, there are AI-related buttons: 'Generate Code with AI' (Ctrl + I) and 'Start with a code example'.

A screenshot of Google Colab. At the top, it says 'Untitled0.ipynb'. The menu bar includes File, Edit, View, Insert, Runtime, Tools, and Help. Below the menu is a toolbar with 'Table of contents', '+ Code', '+ Text', 'Section', and a search icon. The main workspace shows a code cell with the following content:

```
x = 5
x * 5
```



CO TO JE PYTHON?

- Jednoduchost
- Čitelnost kódu
- Rozšiřitelnost
- Univerzálnost
- Dynamické proměnné

```
Python ▾
Copy Caption ...
# Importování balíčků
import numpy as np
import pandas as pd

# Definice funkce
def greet(name):
    print("Ahoj, " + name + "! Vítej v Pythonu.")

# Volání funkce
greet("Jan")

# Definice proměnné
x = 5
y = 3.14
z = "Hello, world!"

# Výpis hodnot proměnných
print(x)

# Použití funkcí z balíčků pandas a numpy
data = np.array([1, 2, 3, 4, 5])
df = pd.DataFrame(data, columns=["Numbers"])
print(df)
```

PROMĚNNÉ

- Python podporuje několik typů proměnných, včetně:
- Celé číslo (Integer): celá čísla bez desetinné části, např. 5, -3, 0
- Desetinné číslo (Float): čísla s desetinnou částí, např. 3.14, -0.5, 2.0
- Řetězec (String): posloupnost znaků, např. "hello", 'world'
- Boolean: buď True nebo False

Úkoly pro trénink proměnných v Pythonu

1. Vytvořte proměnnou s názvem name a přiřadte jí svoje jméno.
2. Vytvořte proměnnou s názvem age a přiřadte jí svůj věk.
3. Vytvořte proměnnou s názvem is_student a přiřadte jí boolean hodnotu, která indikuje, zda jste studentem nebo ne.
4. Vytvořte proměnnou s názvem favorite_number a přiřadte jí svoje oblíbené číslo.
5. Vytvořte proměnnou s názvem gpa a přiřadte jí svoje průměrné známky.

FUNKCE

Python ▾

```
def greet(name):
    print("Ahoj, " + name + "! Vítej v Pythonu.")

# Volání funkce
greet("Jan")
```

Copy

Caption

...

Python ▾

```
def add_numbers(a, b):
    return a + b

# Volání funkce a uložení návratové hodnoty
result = add_numbers(5, 3)
print(result)
```

Copy

Caption

...

FUNKCE

- `len()`: Vrátí délku řetězce.
- `str()`: Převede objekt na řetězec.
- `upper()`: Převede řetězec na velká písmena.
- `lower()`: Převede řetězec na malá písmena.
- `capitalize()`: Převede první písmeno řetězce na velké.
- `strip()`: Odebere mezery na začátku a konci řetězce.
- `split()`: Rozdělí řetězec na seznam podřetězců na základě daného oddělovače.
- `join()`: Spojí prvky seznamu do jednoho řetězce s daným oddělovačem.

LISTS

Ukládají více informací stejného typu

- **seznamy**
- **průběžné výsledky**

- `len()`: Vrátí délku seznamu.
- `append()`: Přidá prvek na konec seznamu.
- `remove()`: Odebere první výskyt daného prvku ze seznamu.
- `pop()`: Odebere a vrátí poslední prvek ze seznamu.
- `insert()`: Vloží prvek na danou pozici v seznamu.
- `index()`: Vrátí index prvního výskytu daného prvku ve seznamu.
- `count()`: Vrátí počet výskytů daného prvku ve seznamu.
- `sort()`: Seřadí prvky seznamu vzestupně.
- `reverse()`: Obrátí pořadí prvků v seznamu.
- `extend()`: Přidá prvky z jiného seznamu na konec seznamu.

Python

```
# Creating a list
fruits = ["apple", "banana", "orange"]

# Accessing elements in a list
print(fruits[0]) # Output: apple

# Adding elements to a list
fruits.append("grape") # Adds "grape" to the end of the list
print(fruits) # Output: ["apple", "banana", "orange", "grape"]

# Removing elements from a list
fruits.remove("banana") # Removes "banana" from the list
print(fruits) # Output: ["apple", "orange", "grape"]

# Modifying elements in a list
fruits[0] = "kiwi" # Replaces "apple" with "kiwi"
print(fruits) # Output: ["kiwi", "orange", "grape"]
```

DICTIONARIES

```
# Creating a dictionary
psychopy_results = {
    "participant_id": "001",
    "response_time": 5.2,
    "accuracy": True,
    "stimulus_type": "visual",
    "experiment_condition": "control"
}

# Accessing values in a dictionary
print(psychopy_results["participant_id"]) # Output: 001

# Adding or modifying key-value pairs in a dictionary
psychopy_results["response_time"] = 6.1 # Modifies the value of an existing key
psychopy_results["experimental_group"] = "treatment" # Adds a new key-value pair
print(psychopy_results) # Output: {"participant_id": "001", "response_time": 6.1, "accuracy": T
rue, "stimulus_type": "visual", "experiment_condition": "control", "experimental_group": "treatm
ent"}

# Removing key-value pairs from a dictionary
del psychopy_results["stimulus_type"] # Removes the "stimulus_type" key-value pair
print(psychopy_results) # Output: {"participant_id": "001", "response_time": 6.1, "accuracy": T
rue, "experiment_condition": "control", "experimental_group": "treatment"}
```

- `keys()`: Vrátí seznam klíčů ve slovníku.
- `values()`: Vrátí seznam hodnot ve slovníku.
- `items()`: Vrátí seznam klíč-hodnota dvojic ve slovníku.
- `get()`: Vrátí hodnotu příslušející danému klíči. Pokud klíč neexistuje, můžete nastavit výchozí hodnotu.
- `pop()`: Odebere a vrátí hodnotu příslušející danému klíči ze slovníku.
- `update()`: Aktualizuje slovník přidáním nebo změnou klíč-hodnota páru ze zadaného slovníku.
- `clear()`: Odebere všechny klíče a hodnoty ze slovníku a vrátí ho prázdný.

ÚLOHY

1. Vytvořte prázdný seznam s názvem `participants`.
 2. Přidejte do seznamu `participants` identifikátory účastníků, například "001", "002", "003".
 3. Vypište prvního účastníka ze seznamu `participants`.
 4. Odstraňte ze seznamu `participants` účastníka s identifikátorem "002".
 5. Upravte identifikátor prvního účastníka ze seznamu `participants` na "999".
-
1. Vytvořte prázdný slovník s názvem `participant_data`.
 2. Přidejte do slovníku `participant_data` klíče a hodnoty, jako je "participant_id" s hodnotou "001", "response_time" s hodnotou 5.2 a "accuracy" s hodnotou True.
 3. Vypište hodnotu klíče "participant_id" ze slovníku `participant_data`.
 4. Přidejte do slovníku `participant_data` klíč "experimental_condition" s hodnotou "control".
 5. Odstraňte ze slovníku `participant_data` klíč "accuracy".

DYNAMYCKÉ VLASTNOSTI

Jak modifikovat stimulus v průběhu trialu?

Cokoli má v názvu \$ je proměnná (python funkce)

Nastavení jednotlivých trialů v Cyklu můžeme randomizovat, objekty se mohou postouvat a měnit barvu

Size [w,h] \$ \$(0.5, 0.5)

Position [x,y] \$ \$(0, 0)

Spatial units from exp settings

Anchor center

constant

constant

constant

set every repeat

set every frame

$\$(t*0.25, t*0.25)$

Variujte pozici/barvu stimulu během trialu

můžete použít funkci "sin(t)"

MODRÍ JIŽ VĚDÍ



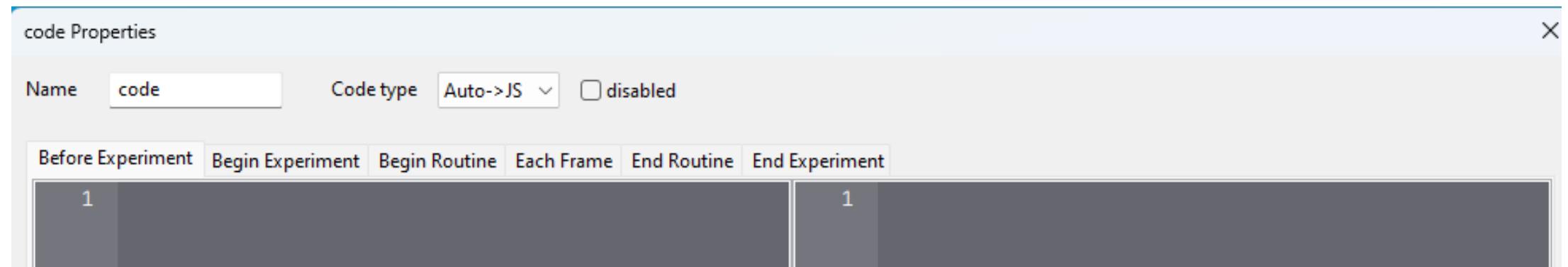
Připravte sérii obrázků

- koček a psů/můžu a žen/cokoli
- Obrázek zamaskujte noise patternem (např. Perlin noise)
- Postupně odhalujte masku (opacity)
- Když participant ví, zvolí vlevo či vpravo



CODE COMPONENT

Jak přidat randomizaci a feedback



PODMÍNKY

Umožňují ukončit, přeskočit trialy na základě odpovědí



The image shows a code editor window with a light gray background. In the top left corner, there is a dropdown menu labeled "Python". On the right side, there is a toolbar with icons for "Copy", "Caption", and three dots. The main area contains the following Python code:

```
x = 5

if x > 0:
    print("x is positive")
elif x < 0:
    print("x is negative")
else:
    print("x is zero")
```

AVAILABLE VARIABLES

Jak přidat randomizaci a feedback

```
clock.getTime()

# Last recorded key response
component_name.properties

# Ends current routine
continueRoutine = False # end the current routine

#ends current loop
trials.finished = True # exit the loop called trials

# the current time (in seconds) measured from the start

# component_name.status (NOT_STARTED, STARTED or FINISHED)
myMovieName.status==FINISHED

# component_name.properties
text_norm.
```

KEYBOARD CONFIRMA TION

Jak reagovat specificky na stisk
klávesy

Python ▾

```
if "e" in keyboard_component_name.keys:  
    print("quitting")  
    endExpNow = True  
  
if "y" in keyboard_component_name.keys:  
    trialReps = 1  
if "n" in keyboard_component_name.keys:  
    trialReps = 0
```

AI OR NOT AI?



Detekce, zdali je
fotografie/text/obraz z
produkce AI či nikoli



GO

NO GO

Welcome screen

Instrukce

20 trialů, z toho 25 procent NOGO
Reaguje mezerníkem - hned na další
Trial je max 2 s dlouhý

Po Trialu přichází Fixační kříž na 0.5 S

Good bye screen

Poznámky

AI OR NOT AI?



[Back to Agenda Page](#)



**STÁHNĚTE/VYGENERUJ
TE MATERIÁL**

Test by měl mít 10 položek



PŘIPRAVTE NASTAVENÍ



PŘIDEJTE FEEDBACK

Briefly elaborate on what you want to discuss.

OKAMŽITÝ

FEEDBACK

Po každé odpovědi zobrazte, zdali byla správně či nešprávně

Obarvěte odpověď zeleně či červeně

přidejte informaci o současném trialu

přidejte informaci o reakčním čae

PRŮBĚŽNÝ

FEEDBACK

Nutné sbírat odpovědi v průběhu

Na feedback obrazovce vyhodnotit počet správných opovědí

Sdělit informaci o počtu zbývajících trialů a procentu správných
odpovědí

Ukončete experiment, pokud se člověk 5krát za sebou netrefí

Na začátku experimentu inicializujte proměnnou počet správných odpovědí

Po správné odpovědi se přičte bod, po špatné nikoli

MULTIPLE SETTINGS FILES

```
$version + ".xlsx"
```