Kísérlet-programozás

Labor - blokkok, visszajelzés és adatkezelés

Blokkok

Egy kísérletben lehet több blokk is. Különböző trial tipusok vagy ugyanazok a trial tipusok egy gyakorló és egy teszt blokkban.

Vegyük a szófelismerés kísérletet az előző óráról.

Tegyünk hozzá egy gyakorló blokkot:

- Legyen egy új Excel (vagy csv) fájl új szavakkal a gyakorláshoz
  - Az oszlopok legyenek ugyanazok, mint a teszt blokkhoz tartozó Excel (vagy csv) fájlban
- Tegyük a trial routine másolatát (Insert Routine) a trial loop elé
- Majd egy új loopot a gyakorló routine köré
  - Adjuk meg a paramétereit
  - A Loop type lehet sequential, hogy megadott sorrendben jöjjenek a gyakorló stimulusok
- Tegyünk a gyakorló loop után egy új routine-t, amiben értesítjük a kísérleti személyt, hogy vége a gyakorlásnak, és gombnyomással továbbmehet a tesztre.

Python kód és visszajelzés

A code tárgy

Code tárgy Python kóddal: jobbról a Custom komponensek között

- Begin experiment fül: ha ide írunk be parancsot, azt a kísérlet indulásakor fogja végrehajtani a program
- Begin routine fül: az ide írt parancsot a code tárgyat tartalmazó routine indulásakor hajtja végre a program
- Each frame fül: ahányszor a képernyő frissíti magát
- End routine fül: a code tárgyat tartalmazó routine végén
- End experiment: a kísérlet végén

Visszajelzés a code tárggyal

- Két módja van:
  - Minden trial után az adott trial-ról
  - Vagy egy blokk után a blokk trial-ok összegzésével
- Trial visszajelzés:
  - A gyakorló loopon **belül** a trial routine után helyezzünk el egy új routine-t
  - Ebbe tegyünk be egy code tárgyat
  - A felső ablakban adjunk neki nevet
  - Az alsó ablakba lehet írni a kódot

Python kód a trial utáni visszajelzéshez

Azt akarjuk, hogy a code komponens olvassa be a trial eredményét, és ezt mentse el egy üzenetben

A trial eredményéről két adatot ment el a Psychopy: hogy helyes volt-e (x.corr) és a reakció időt (x.rt).

x a trialban a választ kérő komponens (a klaviatúra tárgy) neve

1. A Begin experiment fülön megadjuk az üzenet nevét, és induló paramétereit (üresen indul) - ékezet nem lehet a névben, az ürességet két szimpla idézőjellel rendeljük hozzá:

   uzenet = ‘’

2. A Begin routine fülön leírjuk, hogy milyen információ legyen az üzenetben, ha helyes volt a válasz, és mi legyen, ha nem volt helyes a válasz:

```
if szoValasz.corr:
    uzenet = "Helyes! Ido = %.3f" %(szoValasz.rt)
else:
    uzenet = "Nem jo. Ido = %.3f" %(szoValasz.rt)
```

Ha a választ kérő klaviatúra tárgy neve szoValasz.

Dupla idézőjelben az uzenet szövege van.

A % jel egy változó, amit az idézőjel után definiálunk. Itt a szoValasz komponensben adott válasz reakcióideje.

A % után a .3f azt jelenti, hogy a számot három tizedessel írja ki. (f mint floating point)

A parancsban minden írásjel fontos!

Szöveg tárgy az üzenet megjelenítésére

A code komponenst után helyezzünk egy szöveg tárgyat a visszajelzés routine-ba:

Adjunk neki nevet,

Adjuk meg, hogy mikor jelenjen meg, és mennyi ideig maradjon a képernyőn,

A szöveg ez lesz: $uzenet (ha az “uzenet” nevet definiáltuk a code komponensben)

Set every repeat (hogy minden trial után frissüljön)

KÉSZ!

Python kód a blokkot összegző visszajelzéshez

Legyen egy visszajelzés routine a teszt loop **után**. Az elköszönő routine is jó erre.

Jegyezzük meg a teszt loop nevét és a választ kérő tárgy nevét.

Tegyünk egy code tárgyat a routine elejére.

1. A Begin experiment fülön megadjuk az üzenet nevét, és induló paramétereit (üresen indul) - ékezet nem lehet a névben, az ürességet két idézőjellel rendeljük hozzá:

   visszajel = “”

1. A begin routine fülön definiáljuk a blokkot összegző számokat, például:

   Összesen hány trial volt,
   Ebből hányra válaszolt jól a kísérleti személy,
   Mi volt az átlagos reakcióideje.

   És megadjuk, hogy hogy nézzen ki az üzenet (ékezeteket nem szereti)

   ```
   Hanybol = tesztloopneve.data[“valasztargyneve.corr”].count()
   Hanyjo = tesztloopneve.data[“valasztargyneve.corr”].sum()
   Ido = tesztloopneve.data[“valasztargyneve.rt”].mean()
   visszajel = ‘%i-bol %i volt helyes, atlagos ido: %.3f’ %(Hanybol,Hanyjo,Ido)
   ```

A % egy változó, aminek a három előfordulását azonos sorrendben definiáljuk a parancssor végén.

A % után az i azt mondja, hogy egész szám (integer) formátumban jelenjen meg a szám.

Szöveg tárgy az üzenet megjelenítésére

A code komponens után helyezzünk egy szöveg tárgyat a visszajelzés routine-ba:

Adjunk neki nevet,

Adjuk meg, hogy mikor jelenjen meg, és mennyi ideig maradjon a képernyőn,

A szöveg ez lesz: $visszajel (ha az “visszajel” nevet definiáltuk a code komponensben)

Set every repeat (hogy minden trial után frissüljön)

KÉSZ!

Bonyolultabb designok: Loop a loop-ban

A loop-okat egymásba lehet ágyazni. A program a külső loopot hajtja végre először, azután a belsőt.

Példák:

- Hosszú a kísérlet, és szünetet akarunk beépíteni.
- Többféle kíséreti blokkunk van és ezeket random sorrendben szeretnénk bemutatni (pl. Stroop és fordított Stroop egy kísérletben).
- El akarjuk kerülni, hogy bizonyos stimulusok véletlenül egymás után következzenek.

Megvalósítás:

- Minden blokkhoz legyen egy külön Excel vagy csv fájl a stimulusokkal
- Kell még egy Excel vagy csv fájl, amiben a blokkokhoz tartozó Excel fájlok vannak felsorolva

Tárgyfelismerés kísérlet

Alló helyzetben és fejjel lefelé mutatunk tárgyakat. A feladat a tárgyak azonosítása. Lelassul a reakció idő, ha fejjel lefelé látjuk a tárgyat?

Minden ksz mindegyik tárgyat látja álló helyzetben is és fejjel lefelé is. De szeretnénk elkerülni, hogy közvetlenül egymás után lássa ugyanazt a tárgyat a két helyzetben. A ksz-ek fele először fejjel lefelé és aztán állva látja a tárgyakat, a ksz-ek másik fele fordított sorrendben.

Két blokkunk lesz:

- Blokk A: tárgyak álló helyzetben
- Blokk B: tárgyak fejjel lefelé (180 fokkal elfordítva)
- Legyen három Excel fájl:
    1. BlokkA nevű, oszlopai: targy, orientacio, helyesValasz
       A tárgy oszlopa kerülnek a kép fájlok nevei, az orientacio oszlopba 0
    2. BlokkB nevű Excel fájlban ugyanez az információ, de az orientaci oszlo: 180
    3. A harmadik Excel fájlban egy oszlop van: Blokk, ennek két sora van: BlokkA.csv és BlokkB.csv

Amire figyelni kell

- A kép fájlok legyenek a kísérlet mappájában.
  - az images_blocks nevű demo kísérlethez tartoznak képek, ha nincs más
  - A kurzus honlapjáról is le lehet tölteni négy képet
- A fájlok és oszlopok neveiben ne legyen ékezet vagy szóköz
- A kép fájlok neveit pontosan, kiterjesztéssel együtt kell megadni.
- A harmadik Excel vagy csv fájlban a másik két fájl nevét szintén kiterjesztéssel együtt kell megadni. xlsx, xls vagy csv
- Mindhárom Excel fájl legyen a kísérlet mappájában

A kísérlet flow

A routinok:

Bevezetés: szöveg az instrukciókkal, és klaviatúra vagy egér válasz a folytatáshoz

Blokk bevezetés: valami szöveg

Trial:

- kép stimulus: Image $targy (vagy ami a kép nevét tartalmazó oszlop neve), set every repeat
  - Duration üres
  - Size: ki kell próbálni, hogy mi a jó méret. 0.5, 0.5 jó kiindulópont
  - Orientation $orientacio (vagy ami az orientációt tartalmazó oszlop neve), set every repeat
  - Units height (ez az ablak méretéhez igazítja a képet)
- Válasz: klaviatúra a lehetséges gombokkal

Elköszönés

Kísérlet flow

A loop-ok:

- Egy loop a blokk bevezeto és a trial köré
  - Ez a harmadik Excel fájlt hívja be
  - Ha a sorrendet random-ra állítjuk, véletlenszerűen fog választani a két blokk közül - így a ksz-ek fele az A blokkot fogja látni először, a másik fele pedig a B blokkot.
- Egy loop a trial köré
  - Ez a külső loopban behívott harmadik Excel fájl oszlopát hívja be: $Blokk, ami felsorolja a blokkokhoz tartozó Excel fájlokat
  - Ha a sorrendet randomra állítjuk, véletlenszerűen fog választani a blokkon belül a stimulusok közül - mindenki más sorrenben fogja látni a képeket

Kísérlet beállításai

Neve

Milyen információt kérünk a ksz-ektől

- ha a participant szó helyett mást szeretnénk, az output fájl nevében is át kell írni a szót
- Alapértelmezett beállítás: u'data/%s_%s_%s' % (expInfo['participant'], expName, expInfo['date'])
- A % egy változó, az s szöveg formátumot jelöl (s mint string)

Milyen adatfájlokat kérünk

Milyen színű legyen a háttér

Adatfájlok összesítése

A Psychopy egyénenként menti el az adatokat.

Az elemzéshez a csv fájlokat összesíteni kell.

Rettenetesen hosszú és kiborító módszer: bemásolgatni az összes fájlt egybe Excelben

Jobb módszer:

- Az összesítendő .csv fájlokat egy mappába tesszük, ahol nincs más .csv fálj
- Megnyitjuk az operációs rendszer parancssorát
  Windows > run,
  Mac > terminal
- A .csv fájlokat tartalmazó mappába navigálunk: cd parancs mindkét operációs rendszerben

A fájlok mappájából:

- Windows command prompt (run > cmd)
  copy *.csv mergedfile.csv (oszlopnevekkel együtt)
  Remove duplicates az Excel-ből

- Mac Terminal
  Az oszlopneveket egy kivétellel minden fájlról leszedjük, és összesítjük a fájlokat: awk 'FNR > 1' noheaders*.csv > bigfileWithNoHeaders.csv
  noheaders*.csv a lefejezendő fájlok nevei, a * bármilyen karaktersort helyettesít
  aztán egyesítjük a fejetlen fájlt a fejessel: cat megmaradtfájl.csv > mergedfile.csv

Legegyszerűbb módszer bármelyik operációs rendszerre:

- mergecsv.py Python scriptet betesszük a kísérlet mappájába (a data mappán kívül).
- Futtatás: terminal vagy command window cd a .py script mappájába,
- és: python ./mergecsv.py - a data mappában levő összes csv fájlt összefésüli
- Az eredmény a mergedfile.csv nevű fáj lesz, ami mindent tartalmaz, de az oszlopnevek csak egyszer szerepelnek

Még ki lehet próbálni

- Datamerger.ext Window-ra
- Datamerger Mac-re

Adatelemzés

Az összesített fájlt nyissuk meg Excelben vagy Google Sheeten. Libre Office is jó lehet.

Keressük meg a Pivot (angolul) illetve Kimutatás (magyarul) funkciót. Lehet a Data vagy az Insert (Beillesztés) menüben.

Válasszuk ki a teljes adattáblát, és csináljunk belőle Kimutatást egy másik fülön.

A kimutatás összesíti egyénenként az azonos kondícióba tartozó trialek eredményeit:

- A ksz-ek legyenek a sorokban
- A kondíciók legyenek az oszlopokban
- És az accuracy vagy az RT átlaga az adat