

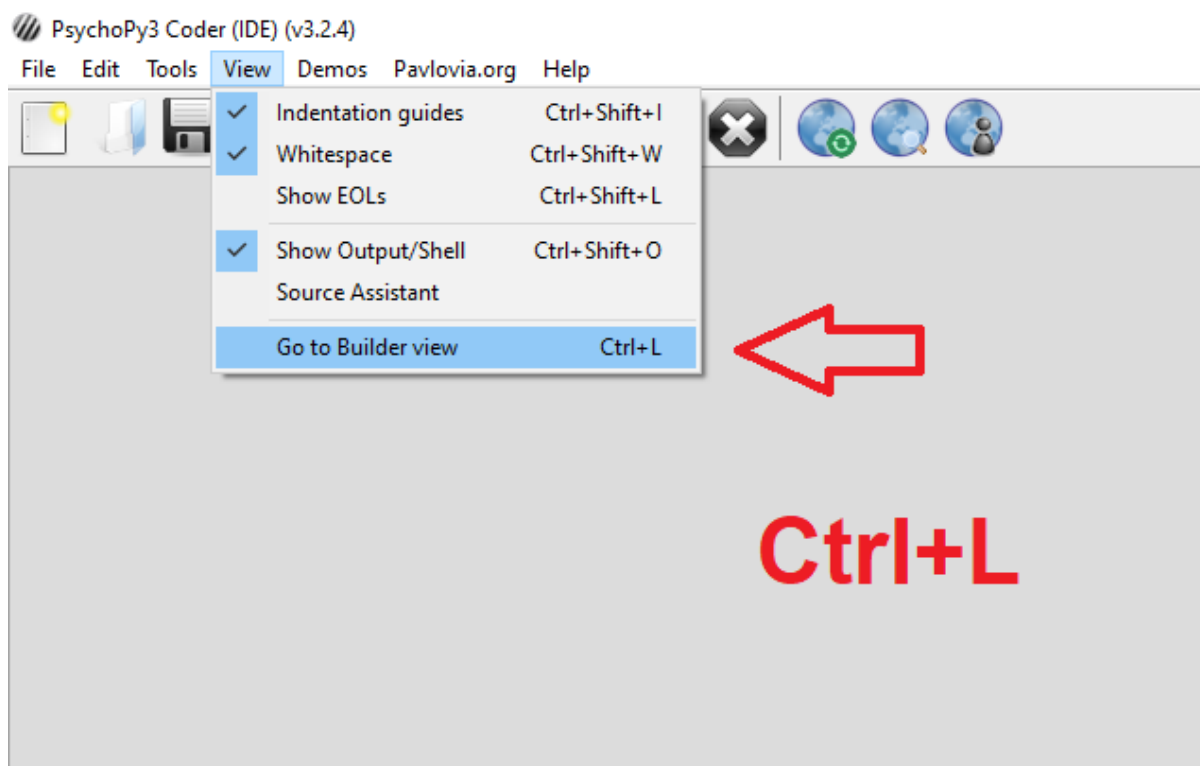
PRAKTIKUM 1: ESIMENE EKSPERIMENT

Eksperimenti juhtfailide loomine pole „raketiteadus“! Käesoleva praktikumi eesmärgiks on muuhulgas demonstreerida, et eksperimenti loomine ei pea olema pikk ja vaevaline protsess, millega on oodatud tegelema vaid üksikud „tehniliselt taibukamad“ inimesed. Tegelikult on eksperimentide loomine sarnane kokandusega. Michelini vääriliste roogade valmistamine on mõistagi terve teadus, kuid ilmselt kahtlustaksime puuduvat motivatsiooni või äärmist tagasihoidlikkust, kui kuuleksime kedagi väitvat ennast kuuluvat inimeste hulka, kes absoluutselt süüa teha ei oska.

Selles praktikumis loome katse, mille eesmärgiks on mõõta klassikalist Stroopi efekti (Stroop, 1935). Täpsemalt vaatame, kuidas üksikut katseseeriat disainida ja katset üles ehitada (sh kuidas stiimuleid esitada ja vastuseid salvestada). Vaatame, kuidas esitused soovitud arvuks kordadeks korduma panna ja kuidas neid igal kordusel eelnevalt defineeritud viisil muutuma panna.

1. PsychoPy graafiline liides ja selle komponendid

Kui avad PsychoPy ja graafiline liides ise ei avanenud, siis vali rippmenüüst: [View > Builder](#). Alternatiivina võid kasutada ka klaviatuuri otseteed **Ctrl + L** (või **Cmd-L** Macintoshi arvutil).



1. Graafilise vaate avamiseks vali punase noolega näidatud valik rippmenüüst või kasuta klaviatuuri otseteed **Ctrl + L** (Mac: **Cmd - L**).

1.1 Komponentid (*components*), rutiinid (*routines*) ja katseskeem (*flow*)

Graafilise vaate paremal ääres on komponentide menüü. Komponentid on eksperimenti ehituskivid. PsychoPy jaotab komponendid funktsiooni alusel nelja suuremasse rühma

(stiimulid, vastused, tellitavad komponendid ja väljundsignaaliga seonduvad komponendid ehk vastavalt *stimuli*, *responses*, *custom* ja *I/O*).

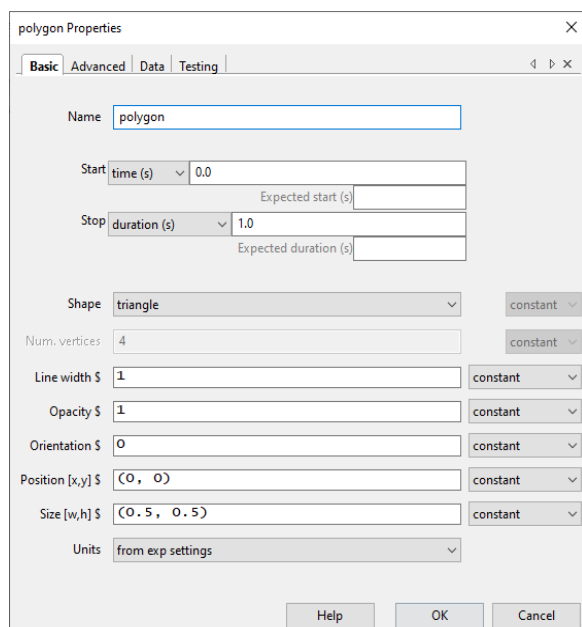
Tüüpilisteks komponentideks võivad olla näiteks esitatavad stiimulid (sh tekstid, pildid, heli) või vastamisega seotud komponendid nagu hiire või klaviatuuri vastused. See, kui paljud neist komponentidest konkreetse eksperimendi ülesseadmisel käiku lähevad, sõltub paljuski konkreetsest eksperimendist. Mõnes eksperimendis võib olla vajadus erinevad komponendid ka omavahel suhtlema panna (näiteks kui teise sündmuse toimumine sõltub esimesest sündmusest, kuid esimese sündmuse toimumise aeg sõltub mõnest juhuslikust tegurist).

Kui tead ette, et mõnd komponenti läheb enam tarvis kui teist, siis võid selle lemmikute alla lisada. Selleks vii kursor valitud komponendi peale ja vajuta paremat hiireklahvi. Vajuta avanenud nupule (*Add to favorites*) ja komponent muutub lemmikute alt kättesaadavaks.

Komponentide klõpsamisel avanevad sätted, milles olevaid muutujaid saad hõlpsasti muuta. Alumisel joonisel on näide visuaalse kujundiga (*polygon*) seotud sätetest. Kuna ühel esitusel võib olla mitmeid komponente (sh teisi visuaalseid kujundeid), siis on soovitatav anda komponendile nimetus, mis peegeldaks tema rolli esituses (nt fikatsioonirist vms). Sätted lubavad muuhulgas määrata komponendi kestuse, suuruse, värvi jpm. Mõningaid nendest võimalustest kasutame ka selles praktikumis. Praktikumis loodavas



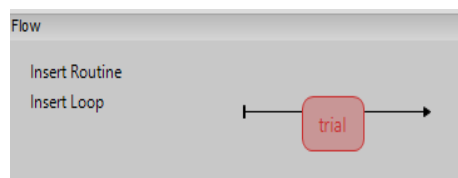
2. Komponentide menüü



4. Näide visuaalse kujundiga (*polygon*) seotud

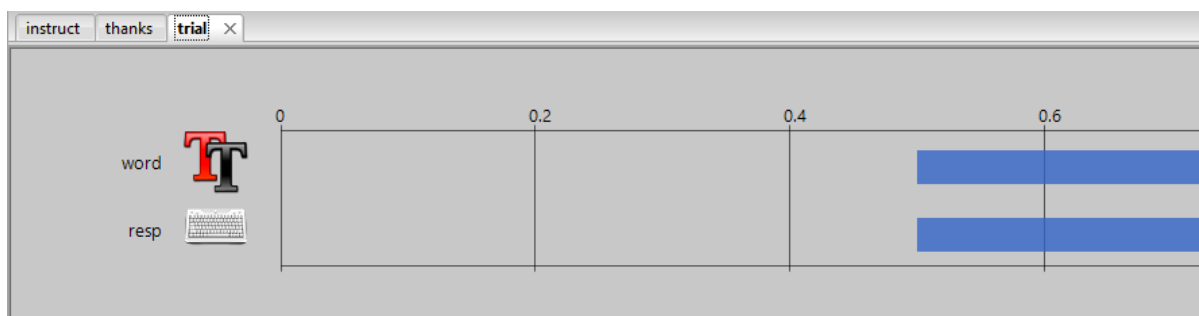
Stroopi katses muudame näiteks tekstikomponendi kestust, sisu ja värvi. Lisaks paneme sisu ja värvi meie poolt defineeritud tingimustel varieeruma.

Sätete menüü all paiknevale „OK“ nupule vajutamisel liidetakse komponent aktiivse rutiinielemendi koosseisu. Vaikimisi on alguses eksperimendil üks rutiinielement, mis kannab nimetust „*trial*“¹.



3. Alguses on eksperimendil üks rutiinielement: „*trial*“.

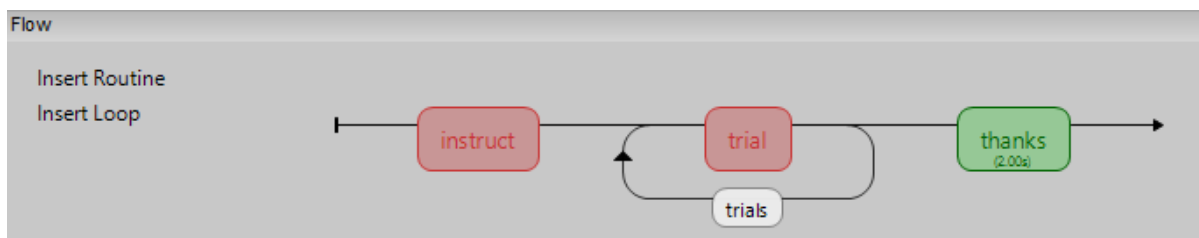
¹ Sõna *trial* on siin (ja edaspidi) jutumärkides rõhutamaks, et tegu on rutiinielemendi nimetusega. Meie poolt loodavas Stroopi katses hakkab trial-i nimeline rutiinielement tähistama katseesitust, kuid võiksime sinna samasse rolli panna ka mõne teise enda poolt loodud rutiinielemendi või muuta olemasoleva rutiinielemendi nime ja rutiinielemendi funktsionaalsus sellest ei muutuks.



5. Stroopi katse esituse näide. Sinised ribad tähistavad hetke, mil konkreetne komponent sisse lülitatakse. Kumbki element lülitatakse sisse 0,5 sekundi pärast „trial“ rutiini algust, kuid nende kestus pole piiratud (sinised jooned ei mahu esitatud ajaskaalale), sest esituse kestus sõltub katseisiku antud vastusest.

1.2 Katseskeem ehk flow

Katseskeemi (*flow*) menüüs määratakse rutiinielementide ajaline järjestus. Skeemi loetakse vasakult paremale ja igast rutiinielemendist võib mõelda kui komponentide salvest (või komponentide kogumist). Rutiinielement võib seega koosneda ühest või mitmest komponendist. Kuigi üks katseesitus võib koosneda ka ühest rutiinielemendist, siis meie poolt loodava Stroopi katse esitus koosneb kolmest rutiinielemendist (ehk „instruct“, „trial“ ja „thanks“ rutiinielemendid Joonisel 6). Katseskeemi (*flow*) menüüs saame määrata ka selle, mitu korda iga rutiinielementi korratakse enne järgmise rutiinielemendi juurde liikumist.



6. Planeeritavas Stroopi katses on kolm rutiinielementi: „instruct“, „trial“ ja „thanks“ (mis vastavad instruksioonile, katse esitusele ja tänusõnadele). „Trial“ rutiini ümber on aas, mis tähistab tsüklit (loop) ja lubab kontrollida, mitu korda konkreetset rutiinielementi katses esitatakse. Rutiinielementi, mille ümber aasa pole esitatakse üks kord.

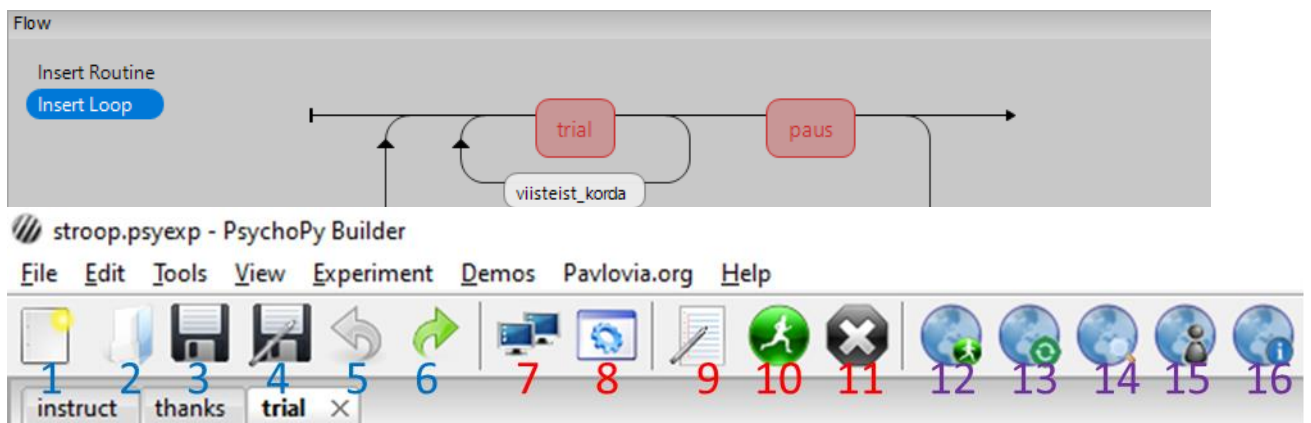
Praktikumis loodavas katses on üks tsükel (*loop*). Tsükel (*loop*) tuleb programmeerimises kasutatavast süntaksist ja neid tsükleid on erinevaid tüüpe nt *while* või *for* tüüpi tsüklid. Tsüklid võimaldavad sama operatsiooni korduvalt ja väikeste muudatustega esile kutsuda selliselt, et poleks vaja sama koodi korduvalt kirjutada. Mitu korda sama asja kirjutamine teeb koodi raskemini jälgitavaks ja võtab ka palju aega. Meie katse näitel tähendaks see, et lisame palju „trial“ elemendi kordusi „instruct“ ja „thanks“ rutiinielementide vahele (kui tahad näha, kuidas graafilises vaates tehtud otsused Pythoni² koodi tasandil peegelduvad vajuta rippmenüü all oleval pliatsi ja paberiga ikoonil). Kui sooviksime esitada vaid mõned seeriad, siis ehk ei tekitakski see väga suurt segadust ja lisatööd (nt juhul kui tahaksime mõnd

² Vaata YouTube'i klippi Pythoni programmeerimiskeele loomisest ja loojast: https://youtu.be/cL_p6zrKQNU

visuaalset illusiooni demonstreerida), kuid suurema arvu korduste korral (nagu psühholoogiliste eksperimentide korral tavaks) on kindlasti praktilisem kasutada tsüklit. See muudab katseskeemi paremini loetavaks ja selle loomine võtab ka vähem aega. Enamikes katsetes on ka vajadus rohkem kui ühe tsükli järele, sest tavaliselt tahaksime teha iga teatud arvu esituste järel pausi, lisada mõned treeningseeriad vms. Näiteks võib meil olla tsükkel, mis ütleb, et rutiinielementi tuleks korrata 15 korda ja selle ümber omakorda teine tsükkel, mis ütleb, et seda tuleks teha 5 korda. Pärast iga kordust võiksime näidata näiteks kirja „paus“ ja nii olekski meie katses kokku 50 kordust ja paus iga viieteistkümne korduse järel.

1.3 Ikoonid ja nende tähendused

Rippmenüü all paikenval ikooniribal olevate mitmete ikoonide funktsioonid on lihtsasti



8. Ikoonide menüü

tuletatavad (nt *New experiment* (1), *Open* (2), *Save* (3), *Undo* (4), *Redo* (5)). Teiste nuppude tähendus võib esialgu olla raskemini tuletatav. Üks võimalus nupu funktsiooni väljaselgitamiseks on sinna peale vajutada ja inspekteerida, mis juhtub. Teine, ja paljudes olukordades optimaalsem (eriti kui nupud hakkavad juba selgeks saama), võimalus on viia kursor nupu kohale, et näha lühikest selgitavat teksti konkreetse nupu funktsiooni kohta.

Monitoride ikooniga nupp (7) võimaldab muuta monitori sätteid (sh resolutsiooni, katseisiku kaugust ja monitori suurust) sellest järgmine mutri pildiga nupp (8) võimaldab muuta eksperimendi sätteid (muuta infodialoogi sisendeid ja sätteid, andmete salvestamist). Pliiatsi ja paberiga nupp (9) muudab eksperimendi Pythoni koodiks, rohelisel taustal oleva valge jooksja siluetiga nupp (10) paneb eksperimendi jooksmas; mustal taustal oleva ristiga nupp (11) paneb programmi seisma (programmi käivitamisel muutub must taust punaseks; NB! stopp nupu vajutamise järgselt (erinevalt *Escape* nupu vajutamisest) andmed ei salvestu).

Nupud 12-16 on seotud katse veebis jooksutamisega või PsychoPy online keskkonnaga üldisemalt. Neist esimene (12) lubab katset veebis demoda, teine (13) sünkroniseerib salvestatud muutused Pavlov.org veebiserveris oleva katsekoodiga, kolmanda (14) nupu kaudu pääseb PsychoPy online katsete keskkonda (kui klõpsad pavlov.org peale ja valid avanenud veebilehe menüüst „explore“, kus näed ja saad katsetada suurt hulka eksperimente ja demosid), neljanda (15) nupu kaudu pääseb PsychoPy gitlabi keskkonda ja viimane nupp (16) annab infot projekti kohta, mille juurde antud fail kuulub (juhul kui fail kuulub mõne online projekti alla).

Ülesanne 0. Loo oma Dokumentide kausta uus kaust, kuhu hakkad loodavaid eksperimendifaile lisama. Pane sellele pealkirjaks PsychoPy_eksperimendid vms. Võid juba luua ka alamkausta, mille nimes oleks sees sõna Stroop (nt minu_Stroop).

2. Stroopi katse

Katse loomisel saame parima tulemuse kui üritame mõelda, millised aspektid jäävad terve katse jooksul samaks ja millised muutuvad esitusest esitusse. See on abiks, sest asjad, mida tahame esitusest esitusse muuta defineerime tingimuste failis, kuid asju, mis jäävad samaks saame kontrollida PsychoPy sees.

Ülesanne 1. Täida tabel. Millised Stroopi eksperimendi aspektid (teksti esitamine, sõna tähendus, sõna värvus, vastuse salvestamine, õige vastus) muutuvad esitusest esitusse ja millised jäävad igal esitusel samaks?

Jääb esitusest esitusse samaks	Muutub igal esitusel

Stroopi katses esitatakse tavaliselt erinevat värvi sõnu, mille tähendus võib kattuda sõna värviga (kongruentne) või mitte kattuda (mittekongruentne). Igal esitusel näidatakse katseisikule sõnu ja registreeritakse vastus, kuid see milliseid sõnu täpselt ja mis värvi peaks saama muutuda igal seerial. Kuna igale värvile vastab erinev nupp klaviatuuril, siis muutub igal seerial ka õige vastus (samamoodi ei muutu vastuse klahvide arv ja klahvide valik).

2.1 Tingimuste faili defineerimine

Nüüd on meil piisavalt infot, et eksperimendi loomisega pihta hakata. Esimese asjana loome tingimuste tabeli. Tingimuste tabel annab loodavale programmile infot, milliseid tunnuseid esitusest esitusse varieerida ja kuidas. Selleks, et programm oskaks tabelit lugeda peaks tingimuste tabel olema kas csv (*comma-separated values*) või xlsx (Microsoft Exceli) formaadis. Tingimuste tabeli loomiseks avame Microsoft Exceli või mõne muu sarnase programmi (nt OpenOffice Calc), mis lubab meil lubatud formaatides andmeid salvestada. Tabeli veergudeks on tunnused (A, B, C, D), mis igal esitusel muutuvad ja read vastavad erinevat tüüpi esitusele. Tehniliselt võttes on veerg D resümee kolmest eelnevast veerust ja seda eksperimendi jooksutamiseks otseselt vaja ei lähe, kuid kuna sellise veeru olemasolu teeb hilisema andmetabeli hõlpsamini jälgitavaks (kõik tabelisse kantud veerud jõuavad ka hilisemasse andmefaili), siis võiksime ta siiski tingimuste tabelisse alles jätta.

Ülesanne 2. Proovi nuputada, mida tabelis toodud veerud tähistavad. Vali järgnevate nimetuste hulgast üks, et anda tabeli veergudele sobivad nimetused: *textColour*, *word*, *corrKey*, *congruent*.

	A	B	C	D
1				
2	red	punane	right	1
3	red	punane	right	1
4	red	roheline	right	0
5	red	sinine	right	0
6	green	roheline	left	1

Ülesanne 3. Hetkel on tabelis viis erinevat esitust/rida. Esimesel neist esitatakse punase värviga (*red*) sõna „punane“, mis vastab paremale nooleklahvile ja viimasel esitusel rohelise värviga (*green*) sõna „roheline“. Õige vastuse saamiseks peaks katseisik viimasel juhul vajutama vasakpoolset nooleklahvi. Lisa tabelisse veel seitse rida selliselt, et punaseid, siniseid ja rohelisi ning kongruentseid ja mittekongruentseid esitusi oleks võrdselt. Salvesta fail oma loodud PsychoPy_eksperimentide Stroopi kausta.

Mõned tähelepanekud, mida veergude nimetamisel silmas pidada:

- Kõigil veergudel peaksid olema nimed (veergude nimed on esimesel real). Nimedeta veerud ajavad PsychoPy segadusse. Üks kavalus on kasutada tingimuste veergude nimetamiseks eesti keelt ning komponentide ja teiste koodielementide nimetamiseks inglise keelt.
- Tabelis ei tohiks esineda kaht sama nimetusega veergu (veergude nimetused ei tohiks kattuda ka ühegi hiljem loodava komponendi nimetusega, sest sellisel juhul muutub ebaselgeks, millal sama nimetus tähistab üht ja millal teist).
- Suur ja väike algustäht: PsychoPy ja täpsemalt Pythoni programmeerimiskeel on tundlik suure ja väikese tähe suhtes. Näiteks on Pythoni jaoks tunnused Word ja word kaks erinevat tühikut (see muutub oluliseks kui hakkame tunnuste tabelit katsesse integreerima).
- Kuna veergude nimetused tõlgitakse Pythoni programmeerimiskeele muutujateks, siis peavad veergude nimetused vastama ASCII formaadile. Kõige lihtsam seda reeglit järgida on kasutada tähti, mis esinevad inglise tähestikus (PsychoPy uuemate versioonide puhul see piirang enam ei kehti).
- Tunnuste nimetused ei tohiks sisaldada tühikuid (ka mitte sõna lõpus) ja kirjavahemärke.

3. PsychoPy: töö komponentidega

Kui tingimuste faili defineerimisel mõtlesime asjadele, mis igal esitusel muutuvad, siis komponentide lisamisel mõtleme asjadele, mis jäävad kõikidel esitustel samaks. Igal esitusel peaks programm suutma esitada sõnu ja registreerima katseisiku vastuse. Kummagi ülesande jaoks on olemas eraldi komponent komponentide menüüs.

Ülesanne 4. Lisa rutiinile „trial“ tekstikomponent ja muuda selle järgmisi prameetreid:

- a) anna sellele uus nimetus (nt stroopText);
- b) määra elemendi algushetkeks 0.5 - ∞;
- c) määra elemendi värvuseks punane;
- d) määra teksti sisuks „punane“.

Ülesande lahendamiseks läheme PsychoPy graafilise liidese aknasse (vt joonist 1).

Veendume, et alguses on meil ees tühi eksperiment. Seda on lihtne kontrollida, sest tühjas eksperimendi failis pole veel komponente ja vaikinisi on seal vaid üks rutiinielement - esitus (*trial*). Salvestame katsefaili tingimuste failiga samasse kausta (...PsychoPy_experimendid/minu Stroop; salvestamise aken avaneb, kui vajutad ikoonile number neli või valid rippmenüüst [File > Save as...](#)).

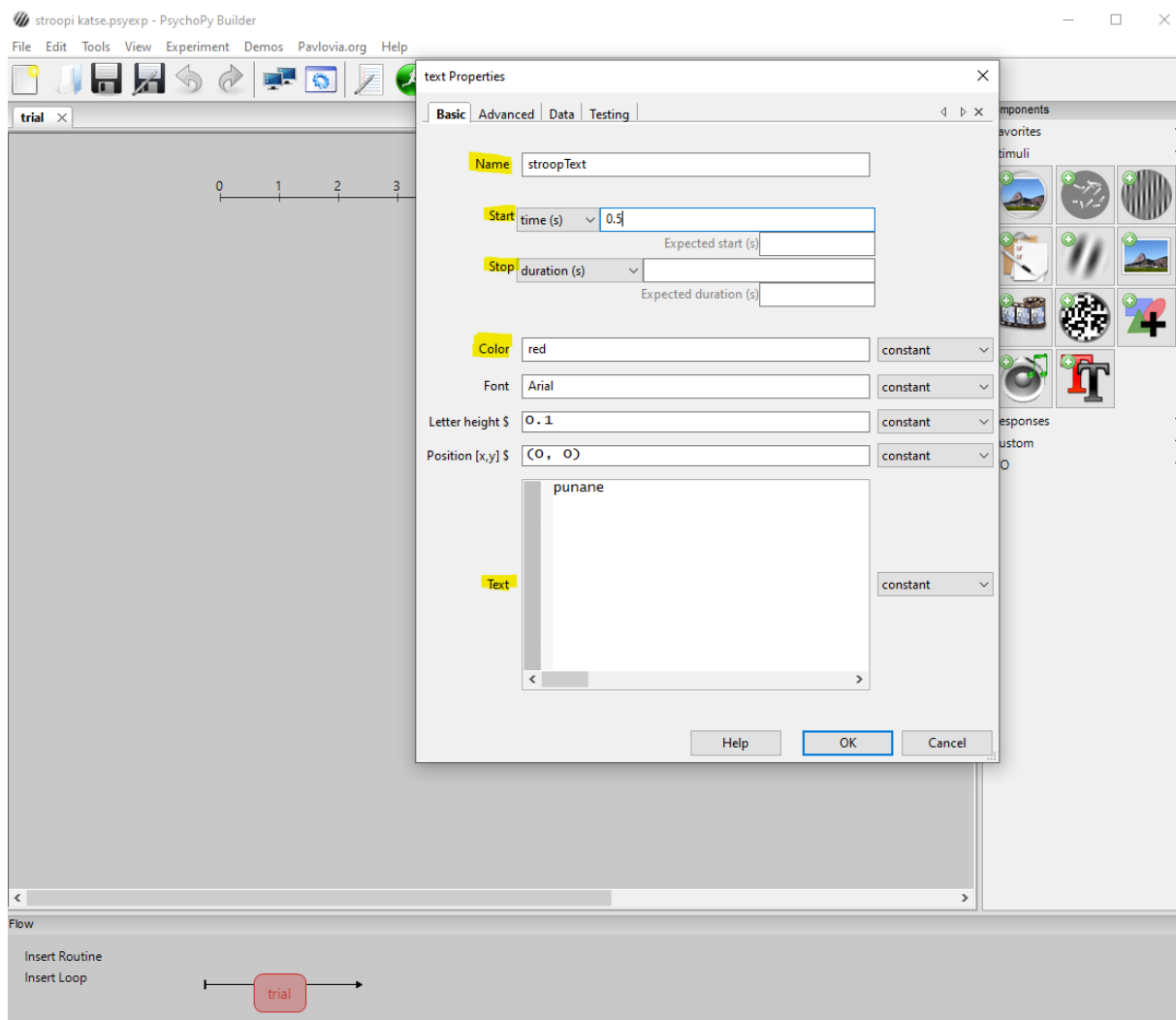
Tekstielemendi lisamiseks valime komponentide menüüst: [Components > Stimuli > Text: present text stimuli](#) (kahe suure T-ga ikoon). Defineeri tekstielemendi nimi (pane tähele, et nimetus ei tohiks kattuda tingimuste faili veergudega), kestus (0.5 sekundit kuni lõpmatus ehk [Start \[time \(s\)\]](#) lahtrisse kirjutame 0.5 ja [Stop \[duration \(s\)\]](#) lahter jääb tühjaks).

Tekstielemendi värvuse punaseks muutmiseks kirjutame [Color](#) lahtrisse *red* ja tekstielemendi sisu defineerimiseks kirjutame lahtrisse [Text](#) „punane“. Kui muudatused on tehtud, siis vajuta „OK“. Seejärel proovi programmi käivitada (programmi käivitamiseks tee hiireklõps rohelisele valge siluetiga ikoonil või kasuta klaviatuuri otseteed [Ctrl + R](#) (Mac: [Cmd - R](#))). Programmi käivitamisel ilmub infokastike. Vaikinisi küsitakse seal katseisiku koodi ja katsekorra või sessiooni numbrit. Antud hetkel ei oma need suurt tähendust, sest tahame ainult kontrollida, kas programm töötab nii nagu vaja.

Praktilisi tähelepanekuid

- Eksperiment ei pea valmima ühe korraga – kasulik on pärast iga suuremat muudatust eksperiment käivitada ja kontrollida, kas kõik ikka töötab plaanilt. Nii on tekkinud vigu ja nende tekkekohti lihtsam ära tunda. Lisaks annab väiksemate sammudega liikumine heaolutunde iga kord, kui programmi täiustamine õnnestub.
- Katse käivitamisel avaneb infokastike. Kuna hetkel pole infokasti vaja, siis võib selle ka ajutiselt välja lülitada. Selleks vajuta mutriga ikoonil (eksperimendi sätted ehk kaheksas ikoon) ja eemalda avanenud aknas linnuke [Show info dialog](#) järelt.
- Kui katse mingil põhjusel edasi ei lähe ja näiteks seisma jääb, siis katse katkestamiseks saab kasutada klaviatuuri vasakus nurgas paiknevat [Esc](#) nuppu.
- Katse käivitamisel avaneb lisaks infokastile veel teinegi „PsychoPy output“ nime kandev aken. Sinna võib PsychoPy printida olulist teavet programmi töö kohta (sh veateateid ja hoiatusi).
- Komponentide sätete menüü avamiseks ja muutmiseks tee komponendil hiireklõps. Komponentide eemaldamiseks vii kursor komponendi peale ja vajuta paremat

hiireklahvi (võibolla pead seda isegi pikemalt kinni hoidma) ning tekkinud menüüst vali *remove*.



9. Tekstikomponendi seadistamine

Sätete tähendused

- Neis näidetes muudame mõningaid levinumaid sätteid, kuid sätete menüüs on veel hulganisti teisigi, millest paljudest tuleb tulevastes praktikumides ka juttu. Kui soovid mõne sätte kohta lisainfot vajuta sätete menüüst „Help“ nuppu. See viib sind PsychoPy kodulehele - seal on komponendi sätteid pikemalt lahti kirjutatud.
- Kui soovid vaid pisikest meeldetuletust konkreetse sätte kohta, siis vii kursor vastava valiku kõrval olevasse lahtrisse ja sinna ilmub täpsustav tekst. Mõnikord võib tekst aidata ka otsustada, millises formaadis programm sisendit ootab. Näiteks kui hiir valiku **Color** sisestuskastile viia ilmub kiri: *Color of this stimulus (e.g. $[1,1,0]$, red); make right click to bring up a color-picker (rgb))*. Nagu kirjeldusest selgub, siis aktsepteerib PsychoPy värvilahter mitut tüüpi sisestusviisi (e.g. $[1,1,0]$, red). Selles praktikumis kasutasime teisena toodud sisestusviisi ehk kirjutasime lahtrisse värvi inglisekeelse nimetuse.

Ülesanne 5. Lisa rutiinielemendile „trial“ klaviatuurivastuse salvestamise komponent ja muuda selle järgmisi seadeid:

- määra vastuste registreerimise algusajaks 0.5 sekundit (sest samal hetkel hakatakse kuvama sõna);
- defineeri lubatud klaviatuurivalikutena järgmised klahvid: vasak, parem ja alla.

Vastuse registreerimise komponendi lisamiseks valime komponentide menüüst (**Components**): **Responses > Keyboard: check and record keypresses** (klaviatuuri pildiga ikoon). Defineeri komponendi nimi (nimetus ei tohiks kattuda nimetustega tingimuste failis ega ühegi teise komponendiga samas katses). Klaviatuurikomponendi algusaja määramiseks kirjuta **Start [time (s)]** lahtrisse algusaeg (sekundites), millest alates peaks klaviatuurikomponent vastuste registreerimist alustama. Kui soovid, et vastamise aeg poleks piiratud, siis tuleks **Stop [duration (s)]** lahter tühjaks jätta. **Allowed keys \$** lahtrisse saame kirjutada klahvide nimetused, mille vajutamist klaviatuurikomponent jälgib.

Kui tahta, et klaviatuurikomponent peaks arvet vasakule, paremale ja alla vajutamiste üle, siis peaksime sinna lahtrisse kirjutama komaga eraldatult ja jutumärkides nende klahvide ingliskeelsed nimetused ('left', 'right', 'down'). Kui sooviksime kasutada tähti, siis kirjutaksime samasse lahtrisse tähed, mille vajutamist tahaksime kontrollida (nt 'a', 's', 'd' kui sooviksime kontrollida tähtede a, s ja d vajutamist).

key_resp Properties

Basic Data Testing

Name: key_resp

Start: time (s) 0.5

Expected start (s):

Stop: duration (s)

Expected duration (s):

Force end of Routine: ☒

Allowed keys \$: 'left', 'right', 'down' constant

Store: last key

Store correct: ☐

Discard previous: ☒

Sync timing with screen: ☒

Help OK Cancel

10. Vastusekomponendi seadistamine

NB!

- Linnuke **Force end of Routine** järel tähendab, et katseisiku nupuvajutus katkestab rutiinielemendi. Vastasel korral ei jõuagi eksperiment järgmiste esituste või muude rutiinielementideni, sest mäletatavasti jätsime elemendi lõppu tähistava lahtri mõlema komponendi puhul tühjaks.
- **Allowed keys** \$ lahtrisse sisestatud nimetused peaksid olema jutumärkides. Jutumärgid on vajalikud, sest vastasel korral ei tea programm, kas pidasime näiteks silmas klahvi *down* või eraldi võetuna tähti d, o, w ja n. PsychoPy pakutud vaikenimetuste ('y', 'n', 'left', 'right', 'space') ümber on üksikud jutumärgid. Pole siiski vahet, kas kasutada topelt või üksikuid jutumärke. Käesoleva dokumendi sees kasutatakse üksikuid jutumärke ainult juhul, kui jutumärgid on nõutavad ka PsychoPy programmis.

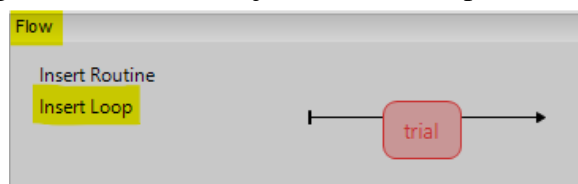
4. PsychoPy: rutiinielementide kordamine ja komponentide värskendamine

Ülesanne 6. Lisa rutiinielemendi „*trial*“ ümber tsükkel (*loop*) ja sea rutiinielemendi korduste arvuks 3.

Tsükkel (*loop*) võimaldab kontrollida mitu korda rutiinielementi korratakse. Tsükli lisamiseks vali katseskeemi menüüst (**Flow**) *Insert Loop*. Seejärel muutub ajajoonel nähtavaks punkt, mille asukohta saab kuni järgmise hiireklahvi vajutuseni muuta (punkti asukoht tähistab tsükli algust või lõppu).

Kui ajajoonel on enam kui üks rutiinielement, siis muutub nähtavaks veel teinegi punkt, mille paigutust saab muuta. Kui ajajoonel on vaid üks rutiinielement, siis fikseerub teine punkt automaatselt.

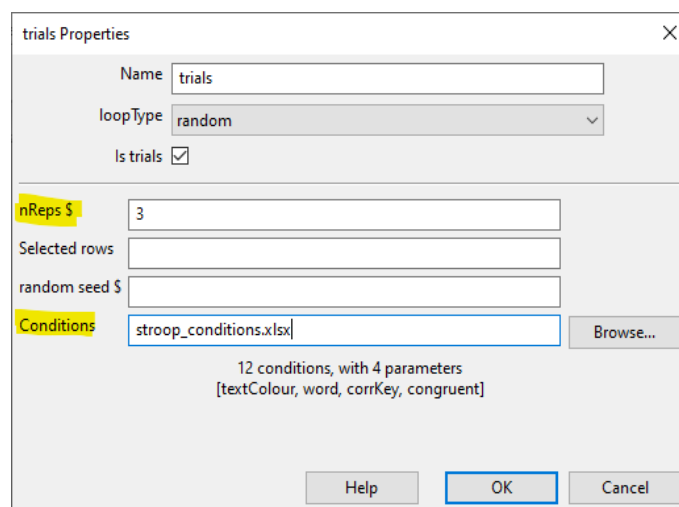
Algus- ja lõpp-punkti valimise järel avanevad sätted. Sarnaselt eelnevalt lisatud komponentidega saab ka tsükli sätteid hiljem hiireklõpsuga nähtavaks muuta või parema hiireklahvivajutusega eemaldada. Soovitud korduste arvu saame kirjutada **nReps** \$ lahtrisse. Proovi katset käivitada ja veendu, et sõna „punane“ esitatakse kolm korda.



11. Rutiinielemendile tsükli lisamine

Ülesanne 7. Lisa tingimuste fail esituse tsüklile.

Tingimuste faili lisamiseks klõpsa hiirega esituse tsükli peal. Avanenud menüüs klõpsa **Conditions** lahtri järel paikneval valikul „*Browse...*“. Avanenud lehitsemisaknas klõpsa tingimuste failil ja vajuta „*Open*“. Aken avab kausta kuhu eksperimendi juhtfaili eelnevalt salvestasid. Kui tingimuste faili lehitsemisaknas ei paista, siis veendu, et tingimuste fail oleks ikka katse juhtfailiga samas kaustas. Kui tingimuste fail sai valitud ja õigesti seadistatud peaksime selle all nägema tingimuste faili kirjeldust. Meie tingimuste failis peaks olema neli tunnust (veergu) ja 12 tingimust (rida).



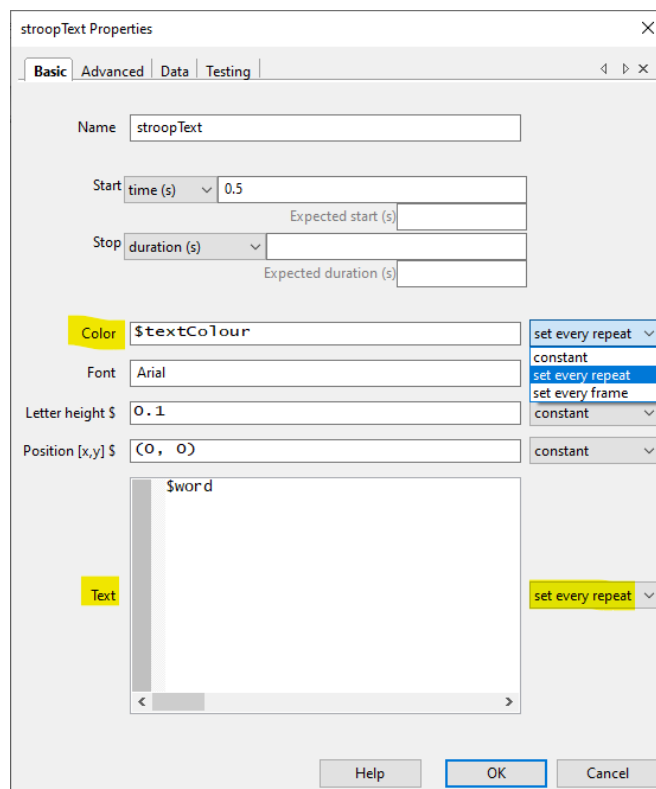
12. Rutiinielemendi kordamisega seotud sätted

Kui enne tähistas **nReps** \$ lahtrisse sisestatud arv üksikute esituste arvu, siis pärast tingimuste faili lisamist tähistab see seda, mitu korda tingimuste faili katses korratakse. Kui jätame arvu muutmata tähendab see, et meie katses on kokku 12×3 ehk 36 üksikesitust. Muidugi tahame ka, et programm ei esitaks meie sisestatud ridu järjest vaid juhuslikult. Selle eest hoolitseb **loopType** järel olev valik *random*. Täpsemalt tähendab see, et programm võtab ette meie tingimuste faili ja ajab selle read segamini. Seejärel hakkab ta sellest segatud tabelist ükshaaval ridu võtma seni, kuni kõik read on läbi tehtud. Nüüd ajab programm read uuesti segamini ja kordab tehtut ning seda kokku **nReps** \$ lahtris määratud korda (antud juhul kolm). **loopType** järel oleva valiku peale klõpsates näed, et valikuid on teisi. Sellest, mida teised valikud tähendab tuleb juttu järgmistes praktikumides.

Ülesanne 8. Muuda tekstikomponendi sätteid selliselt, et teksti sisu ja värvus muutuksid igal esitusel vastavalt tingimuste failile.

Selleks ava tekstikomponendi seaded (tee muutmist vajaval komponendil hiireklõps). Tuleta meelde, mis oli selle tingimuste faili veeru nimi, kuhu sisestasime inglisekeelsed värvinimetused. Kirjuta valiku **Color** järel selle veeru nimi ja veeru nime ette lisa dollari märk (\$), eestikeelsel klaviatuuril **Ctrl + Alt + 4** (või **Shift + 4** Macintoshi arvutil). Dollari märk annab PsychoPy-le märku, et tegu pole teksti vaid lühikese koodijupiga. Lisa ka valiku **Text** lahtrisse veeru nimetus, kuhu eelnevalt eestikeelsed värvinimetused salvestasime (ka siin tuleks PsychoPy-d teavitada, et seda teksti tuleks koodijupina interpreteerida). Muuta tuleks ka kummagi sisestuskasti järel olevat muutujat. Vaikimis on seal *constant*, mis tähendab, et tekstikasti sisestatud väärtus katse jooksul ei muutu ehk jääb konstantseks. Ülejäänud kaks valikut on *set every repeat* ja *set every frame*. Meile sobib neist esimene ehk *set every repeat*, sest soovime, et igal uuel esitusel valiks katseprogramm meie määratud

veerust uue väärtuse. Valik *set every frame* tähendaks, et soovime lahtri sisu pärast iga ekraanivärskendust uuendada³.



13. Tekstikomponendi seadete muutmine

Milles seisneb *set every repeat* ja *set every frame* erinevus?

- Tegelikult toimivad *set every repeat* ja *set every frame* valikud käesoleval juhul väga sarnaselt, sest PsychoPy võtab iga esituse alguses meie poolt lisatud tingimuste faili vastavatest veergudest ühe väärtuse ning seda väärtust (nt „red“) sama esituse jooksul ei muudeta. Siiski pole päris korrektne järeldada, et polegi tähtis kumba kasuks valiku kasuks otsustada. Põhjus on selles, et esimesel juhul ütleme PsychoPy-le, et see muutuja jääb terve esituse jooksul samaks, aga teisel juhul luiskame ja ütleme PsychoPy-le, et pärast iga värskendust võib midagi hakata muutuma. Valik muudab seega programmi valmisolekut sisenditeks, kuid valmisolek ise veel nähtavalt midagi ei muuda. Küll aga muutub see, millistele protsessidele programm arvutusvõimsust broneerib. Kui tahame, et meie loodud katse oleks ajaliselt võimalikult täpne, siis peaksime valima meetodi, mis on käesoleva esitlusviisi seisukohast optimaalseim. Nii

³ Kui sa pole kindel, mida siin „ekraani värskenduse“ all silmas peetakse, siis pole midagi. Sellest tuleb põhjalikumalt juttu tulevases praktikumis. Hetkel piisab kui tead, et arvutimonitori pilt on vaid näiliselt staatiline, kuid tegelikult saavutatakse staatiline mulje üksteisele järgnevate kiirete vilgutuste ehk värskenduste tulemusena. Sellest vilkumissagedusest sõltub, millal ja kui tihti monitorile esitatud pilte saab muuta.

säästame arvutusvõimsust ja saame kindlamad olla selles, et stiimulite esitusajad on lähedased sellele, mida soovisime.

Ülesanne 9. Muuda vastusekomponendi sätteid selliselt, et info õige vastuse kohta muutuks igal esitusel vastavalt tingimuste failile ja katseisiku vastus ka salvestuks.

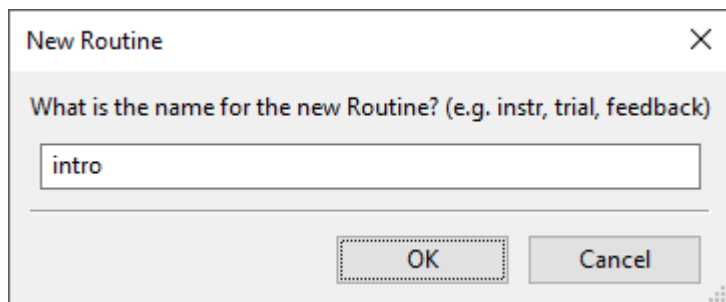
Seadete avamiseks klõpsa hiirega vastusekomponendil. Selleks, et info õige vastuse kohta igal esitusel uueneks kirjutata **Correct answer** lahtrisse selle tingimuste faili veeru nimi, kuhu kirjutatakse klahvide ingliskeelsed nimetused. Sarnaselt tekstikomponendiga tuleks ka siin nime ette dollari märk (\$) lisada. Veendu, et valiku **Store correct** järel on linnuke, sest vastasel korral vastust ei salvestata.

14. Vastusekomponendi seadete muutmine

5. PsychoPy: rutiinielementide lisamine

Ülesanne 10. Lisa katsele kaks uut rutiinielementi. Esimese rutiinielemendi eesmärgiks on katse sisse juhatada. Selles peaksid olema tekstikomponent ja vastusekomponent, et katseisik saaks nupuvajutusega sissejuhatusesest edasi minna ja katse käima panna. Teise rutiinielemendi eesmärk on anda märku, et katse sai läbi ja tänada katseisikut panustatud aja eest. Muuda teise rutiini tekstikomponenti selliselt, et tekst oleks ekraanil 2 sekundit.

Uue rutiinielemendi lisamiseks vali katseskeemi menüüst (Flow) **Insert routine > (new)**.
 Avanenud aknasse kirjuta uue rutiinielemendi nimi (näiteks intro) ja vajuta „OK“.
 Analoogselt tsükli paigutamisele ilmub ka siin ajajoonel must täpp, mis lubab rutiinikomponendile kuni järgmise hiireklõpsuni kohta otsida.



15. Rutiinielemendi nimetamine

Rutiinielementide nimetamine ja eemaldamine

- Rutiinielemendi asukoha vahetamiseks tuleks menüüst sama rutiin uuesti valida ja soovitud kohta asetada. Soovimatu koha peal paikneva rutiinielemendi eemaldamiseks tee rutiinikomponendil parem klõps ja vali *remove*. Parema klõpsuga nähtavaks tulnud menüüs on veel teinegi valik, mis lubab rutiinielemendi nime vahetada (*rename*).
- Kui kasutad mitmes kohas sama rutiini, siis muutub kõigi nende rutiinide nimetus.

Sarnaselt **ülesannetele 4 ja 5** tuleks ka neile rutiinielementidele lisada tekstikomponent ja sissejuhatuse puhul ka vastusekomponent. Mõtle, millised nendes ülesannetes tehtud muudatused on nende rutiinielementide puhul vajaliku ja millised mitte (vihje leiad allpool). Esmalt aga veendu, et oled õige rutiini vaates – vaadet saad muuta rutiinielemendile (näiteks „trial“ või „intro“) klõpsates või menüüriba all samanimelistele tab-idele vajutades. Viimasega tasub ettevaatlik olla, sest tab-i sulgemine kustutab rutiini. Kui sulgesid tab-i kogemata, siis saad *undo* ikoonile (vasakule poole näitav kollane nool) vajutades ta tagasi tuua; sama teeb ka klaviatuuri otsetee Ctrl+Z (Mac: Command-Z). Selline lahendus töötab kahjuks ainult hiljuti tehtud muudatuste tagasipööramiseks.

Sissejuhatava rutiinielemendi...

- a) tekstikomponendiga seotud muudatused:

Stop [duration (s)] lahter jääb tühjaks (et katse ise edasi ei läheks).

Text lahtrisse tuleks sisestada sissejuhatav tekst, näiteks: „Tere tulemast katsessesse! Katse alustamiseks vajuta palun tühikuklahvi...“.

- b) klaviatuuri komponendiga seotud muudatused:

Allowed keys lahtrisse tuleks sisestada lubatud klahvi nimetus (nt 'space' kui soovime, et katseisik saaks tühikuklahvi vajutusega katse käima panna).

Väljajuhatava rutiinielemendi...

a) tekstikomponendiga seotud muudatused:

Stop [duration (s)] lahtrisse sisesta kui kaua soovid, et tekst ekraanil oleks (nt 2 sekundit).

Text lahtrisse tuleks sisestada tekst, mis ütleks, et katse on läbi ja tänaks katseisikut, näiteks: „Katse on lõppenud. Täna, et leidsid aega katses osaleda!“.

6. PsychoPy: infokasti seadistamine

Ülesanne 11. Muuda katse käivitamisel ilmutatavat infokasti selliselt, et sinna saaks sisestada katseisiku numbrit, soo ja vanust.

Katse käivitamisel avaneb infokast. Infokasti sisestatud andmed lisatakse katseisiku andmefaili ja seega on see mugav viis katseisikuga seotud info küsimiseks või salvestamiseks. Infokasti seadistamiseks vajuta mutriga ikoonil. Kui infokast käivitamisel ei avane, siis põhjuseks võib olla see, et **Show info dialoog** järel puudub linnuke. Lahtrite lisamiseks või eemaldamiseks kasuta **Experiment info** sisestuskastide kõrval olevaid miinus ja pluss märke. Esimene sisestuskast annab väljale nime ja teise lahtrisse saab kirjutada vaikeväärtused, mida saab eksperimendi käivitamisel muuta. Täidetavad väljad ei ole

Field	Default
Katseisiku number	
Sugu (m/n/muu)	
Vanus	

16. Infokasti seadete muutmine

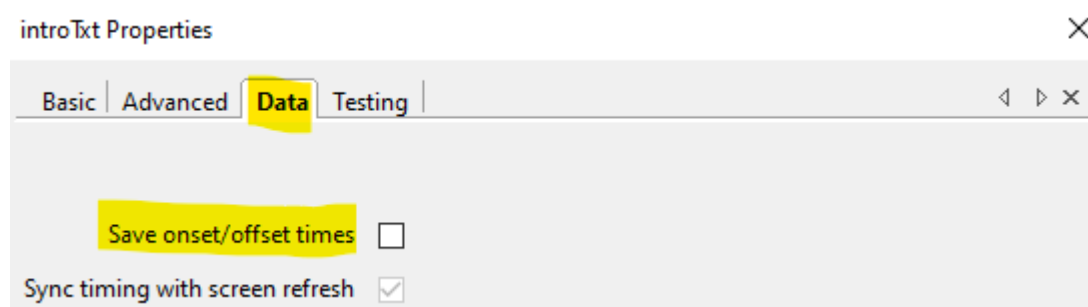
kohustuslikud ja katse käivitub ka siis, kui lahtrid tühjaks jätta. Kui vaikeväärtuseks kirjutada mõni keelatud sümbol (nt küsimärk), siis ei lähe eksperiment edasi enne, kui keelatud sümbol on kustutatud. See võib olla käepärane viis, kuidas end hajameelsuse eest kaitsta ja hilisemat andmetega seotud segadust vältida.

Ülesanne 12. Muuda *trial* tsüklit selliselt, et katses esitataks iga tingimustetabeli veergu ühe korra. Käivita katse ja tee katse ühe korra läbi. Seejärel ava andmefail. Mitu rida on andmefailis ja miks? Veendu, et leiad andmefailist kõik veerud, mille defineerisid tingimuste

failis. Andmefailis peaksid olema ka eraldi veerud infokasti sisestatud info jaoks. Millist kaht veergu kasutaksid kontrollimaks, kas reaktsiooniajad erinesid kongruentses ja mitte-kongruentses tingimuses?

Andmefailis peaks olema 12 unikaalset seeriat. Kui lisaks neile on veel mõni rida, siis võib põhjuseks olla see, et programm salvestas vaikimisi ka aja, mil sissejuhatava ja väljajuhatava rutiini komponendid käivitati.

Kui soovid nende ridade kirjutamise välja lülitada, siis ava üksteise järel nende komponentide seaded. Seadete avamisel tuleb kõigepealt lahti *Basic* vaade ehk vaade, milles oleme seni muudatusi teinud. Komponenti sisselülitamisega seotud aja salvestamise väljalülitamiseks tuleks ülevalt menüüst vali *Data* vaade ja võtta ära linnuke *Save onset/offset times* eest. „OK“ vajutamise järel selle komponendi sisselülitamise kohta infot andmefaili ei salvestata.



17. Komponentiga sisselülitamisega seotud aja salvestamise väljalülitamine

Lisaülesanded

Ülesanne 1. Muuda Stroopi katse tsükli nii, et katse läheks otse sissejuhatuse juurest tänusõnade juurde. Rutiinielementide vahelejätmine on käepärane viis katse funktsionaalsuse testimiseks.

Ülesanne 2. Muuda eksperimenti selliselt, et enne esimest seeriat oleks pisike paus (näiteks 2 sekundit).

Ülesanne 3. Tee lühikene eksperiment ja kontrolli, kuidas erineb Pythoni koodi ridade arv olukorras, kui *set every repeat* asemel on hoopis *set every frame* sisse lülitatud.

Vt ka PsychoPy asutaja (Jonathan Peirce) Stroopi katse loomise õpetust YouTube'is:

<https://www.youtube.com/watch?v=VV6qhuQgsiI>