

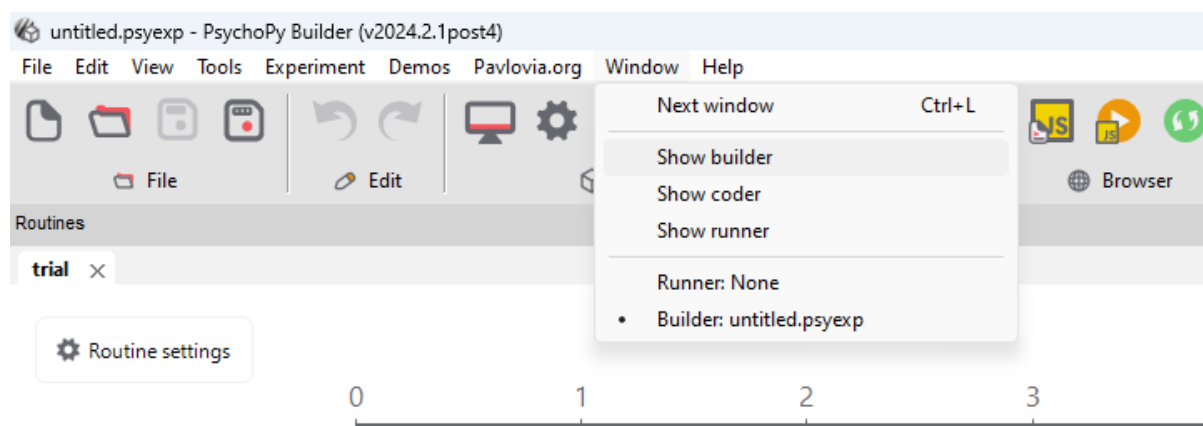
PRAKTIKUM 1: ESIMENE EKSPERIMENT

Eksperimendi juhtfailide loomine pole „raketiteadus“! Käesoleva praktikumi eesmärgiks on muuhulgas demonstreerida, et eksperimendi 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: [Window > Builder](#).



Tavaliselt avaneb PsychoPy käivitamisel kolm akent: 1) Coder (IDE), milles saad käsurealt Pythoni koodi kirjutada ja jooksutada; 2) PsychoPy Builder, milles selle kursuse jaoks valdavalt enda katseid loome ja jooksutame; ning 3) Experiment Runner, mis on aken, kuhu saame enda katsed käivitamist ootama saata ja mis annab meile ka veateateid, kui PsychoPy programmis mõne vea tuvastab.

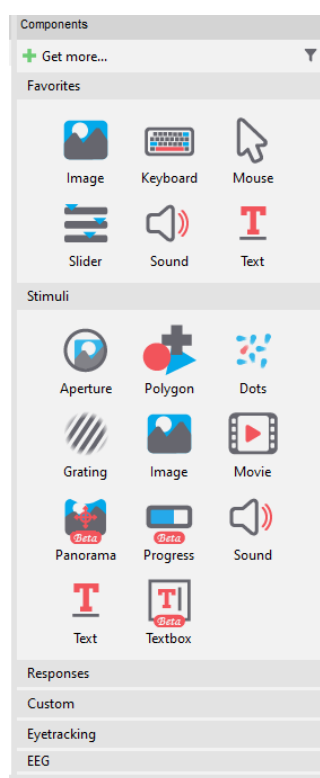
1.1 Komponendid (*components*), rutiinid (*routines*) ja katseskeem (*flow*)

Graafilise vaate paremal ääres on komponentide menüü. Komponendid on eksperimendi ehituskivid. PsychoPy jaotab komponendid funktsiooni alusel rühmadesse (sh stiimulid, vastused, tellitavad komponendid, elektroentsefalograafia ja väljundsignaaliga seonduvad komponendid ehk vastavalt *stimuli*, *responses*, *custom*, *EEG* ja *I/O*).

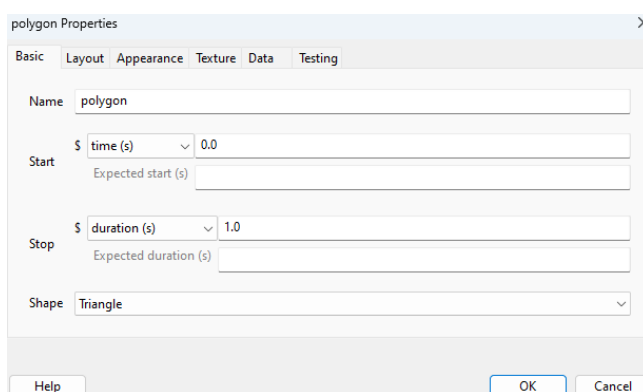
Tüüpilisteks komponentideks võivad olla näiteks esitatavad stiimulid (sh tekstid, pildid, heli) või vastamisega seotud komponendid nagu hiire või klaviatuuri vastused. See, kui paljud neist komponentidest konkreetse eksperimendi ülesseadmisel käiku lähevad, sõltub paljuski

konkreetselt eksperimentidest. Mõnes eksperimentis võib olla vajadus erinevad komponendid ka omavahel suhtlema panna (näiteks kui teise sündmuse toimumine sõltub esimesest sündmusest ja esimese sündmuse kestus pole fikseeritud suurus, vaid näiteks loositakse igal seerial juhuslikult).¹

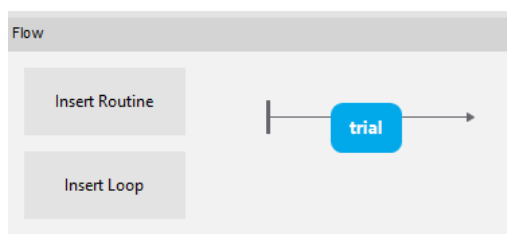
Komponendile klõpsamisel avanevad sätted, mida saad hõlpsasti muuta. Alumisel joonisel on näide visuaalse kujundiga (*polygon*) seotud sätetest. Kuna ühel esitusel võib olla mitmeid komponente (sh teisi visuaalseid kujundeid), siis on soovitatav anda komponendile nimetus, mis peegeldaks tema rolli esituses (nt fikatsioonirist vms). Sätted lubavad muuhulgas määrata komponendi kestuse, suuruse, värvi jpm. Mõningaid nendest võimalustest kasutame ka selles praktikumis. Praktikumis loodavas Stroopi katses muudame näiteks tekstikomponendi kestust, sisu ja värvi. Lisaks paneme sisu ja värvi meie poolt defineeritud tingimustel varieeruma.



Komponentide menüü



Visuaalse kujundiga (*polygon*) seotud sätted



Vaikimisi on igal katseskeemil üks rutiinielement (toodud joonisel sinisega)

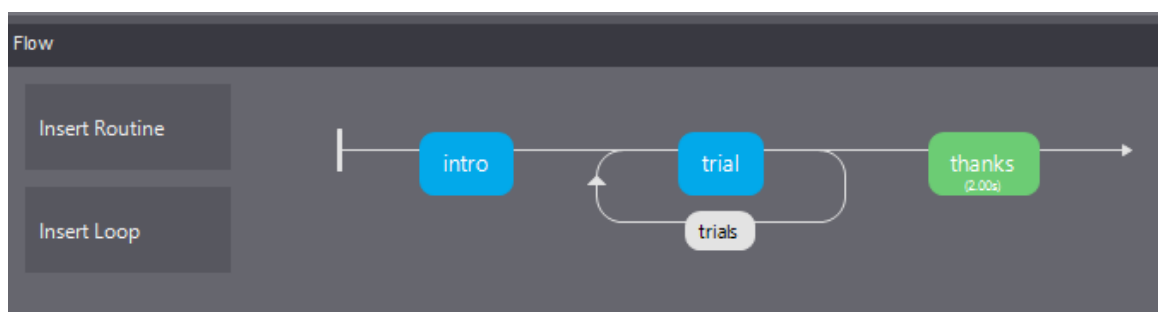
Sätete menüü all paiknevale „OK“ nupule vajutamisel liidetakse komponent aktiivse rutiinielemendi koosseisu. Vaikimisi on alguses eksperimentil üks rutiinielement, mis kannab nimetust *trial*.

1.2 Katseskeem ehk flow

¹ Kui märkad, 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.

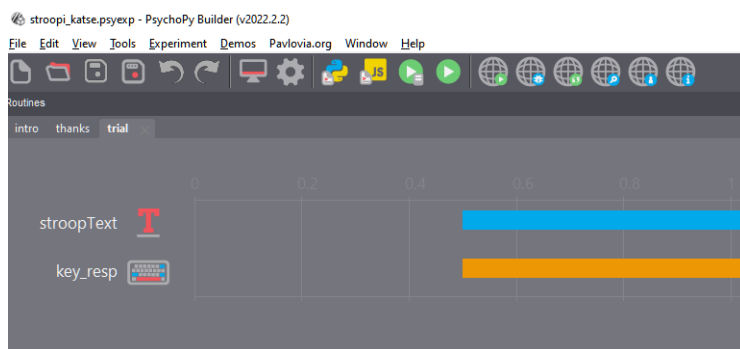
Katseskeemi (*flow*) menüüs määratakse rutiinielementide ajaline järjestus. Skeemi loetakse vasakult paremale ja igast rutiinielemendist võib mõelda kui komponentide kogumist, mis koosneb ühest või mitmest komponendist.

Kuigi katse võib koosneda ka ühest rutiinielemendist, siis meie poolt loodava Stroopi katse on neid kolm: *intro*, *trial* ja *thanks* (vt alumine joonis). Katseskeemi (*flow*) menüüs saame määrata ka selle, mitu korda iga rutiinielementi korratakse enne järgmise rutiinielemendi juurde liikumist. Praktikumis loodavas katse on üks tsükel (*loop*). Tsükli (*loop*) nimetus viitab programmeerimisele ning 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.



Planeeritavas Stroopi katse on kolm rutiinielementi: *intro*, *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 katse esitatakse. Rutiinielementi, mille ümber aasa pole, esitatakse üks kord.

Meie katse näitel saame *trial* rutiinielementi pisikeste muudatustega palju kordi esitada selliselt, et ei pea seda rutiinielementi igaks esituseks eraldi looma (kui tahad näha, kuidas graafilises vaates tehtud otsused Pythoni koodi tasandil peegelduvad, vajuta rippmenüü all oleval pythoni ikoonil ehk sinise ja kollasega ikoonil). Kui sooviksime esitada vaid mõned seeriad, siis ei oleks sama rutiinielemendi kordamine ehk väga suur probleem ja lisatöö (nt juhul kui tahaksime mõnd 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.

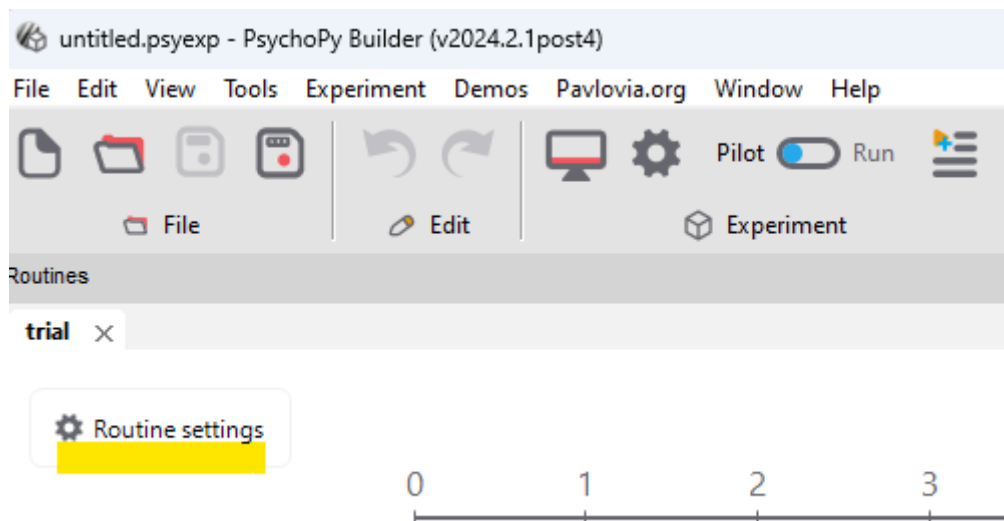


Stroopi katse esituse näide. Värvilised ribad (sinine ja kollane) tähistavad hetke, mil konkreetne komponent sisse lülitatakse. Kumbki element lülitatakse sisse 0,5 sekundit pärast trial rutiini algust, kuid nende kestus pole piiratud (sinised jooned ei mahu esitatud ajaskaalale), sest esituse kestus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

1.3 Ikoonid ja nende tähendused

Rippmenüü all paikenval ikooniribal olevate mitmete ikoonide funktsioonid on varasema kogemuse ja pildi põhjal arvatavasti lihtsasti tuletatavad (nt *New experiment* (**1**), *Open* (**2**), *Save* (**3**), *Save experiment as...* (**4**), *Undo* (**5**), *Redo* (**6**)). 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 infokastikese sisendeid ja sätteid, andmete salvestamist ja mitmeid teisi vaikesätteid). Uuemas PsychoPy versioonis saab eraldi muuta ka iga rutiinielemendi vaikesätteid (vt **Routine settings**).



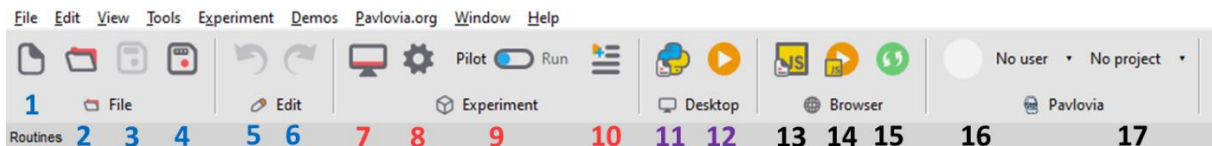
Uuemas versioonis saab muuta ka iga rutiinielemendi vaikesätteid eraldi

Pilot ja *Run* siltidega ikoon (**9**) võimaldab aktiveerida vastavalt, kas piloteerimise või eksperimendi jooksumise režiimi. Teksti ridu meenutava ikooni (**10**) peale vajutamisel saadetakse juhtfail käivitamisaknasse (*Experiment Runner*), kuid juhtfaili ennast veel ei käivitata.

Sinise ja kollasega Pythoni ikooniga nupp (**11**) muudab eksperimendi Pythoni koodiks ja tumekollasel taustal olev esimene valge kolmnurgaga nupp (**12**) käivitab eksperimendi tavalisel moel. Piloteerimise režiimis kuvatakse eksperiment väikesel ekraanil ja ka käivitamisaknasse prinditakse selles olekus rohkem infot, et omaksime paremat ülevaadet juhtfaili toimimise kohta ja saaksime vajadusel muudatusi teha.

JS kirja kandev nupp (**13**) tõlgib eksperimendi Javascripti programmeerimiskeelde ja sellest järgmine tumekollasel taustal olev valge kolmnurgaga nupp (**14**) käivitab programmi veebis. Piloteerimise režiimis käivitub katse küll veebilehitsejas, kuid ei nõua veel programmi enda veebi laadimist, mis on käepärane viis veebikatsete piloteerimiseks. Veebikatsete loomist

vaatame kolmandas praktikumis ja järgnevad valikud ongi seotud sellega. Kahe valge noolekesega roheline nupp (15) laeb eksperimendi või viimasena tehtud muudatused veebi, kuid selle toimimiseks tuleks esmalt Pavloviasse sisse logida (16). Pärast sisselogimist viib ikoonile vajutamine otse Pavlovia keskkonda, kus saki „Experiments“ all on nähtavad kõik sisse logitud kasutaja katsed/projektid. Viimane ikoon (17) näitab, millist nimetust veebikatses kaust Pavlovia kannab ja võimaldab veebikatses tarbeks uue kausta luua (sama funktsioon käivitub ka sellisel juhul, kui esimest korda valge noolekesega rohelisele nupule vajutata).



Ikoonide menüü

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 asjad, mis jäävad esitusest esitusse samaks kontrollitakse 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 esitustele. 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: *textCol*, *word*, *corrKey*, *congruency*.

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

Ülesanne 3. Hetkel on tabelis viis erinevat esitust/rida. Esimesel neist esitatakse punase värviga (*red*) sõna „roheline“, mis vastab vasakule nooleklahvile ja viimasel esitusel rohelise värviga (*green*) sõna „sinine“. Õige vastuse saamiseks peaks katseisik viimasel juhul vajutama alumist 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). Nimetamata veerud ajavad PsychoPy segadusse.
- 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 tunnust (see muutub oluliseks kui hakkame tunnuste tabelit katsesse integreerima).
- Tunnuste nimetused ei tohiks sisaldada tühikuid (ka mitte sõna lõpus) ja kirjavahemärke.

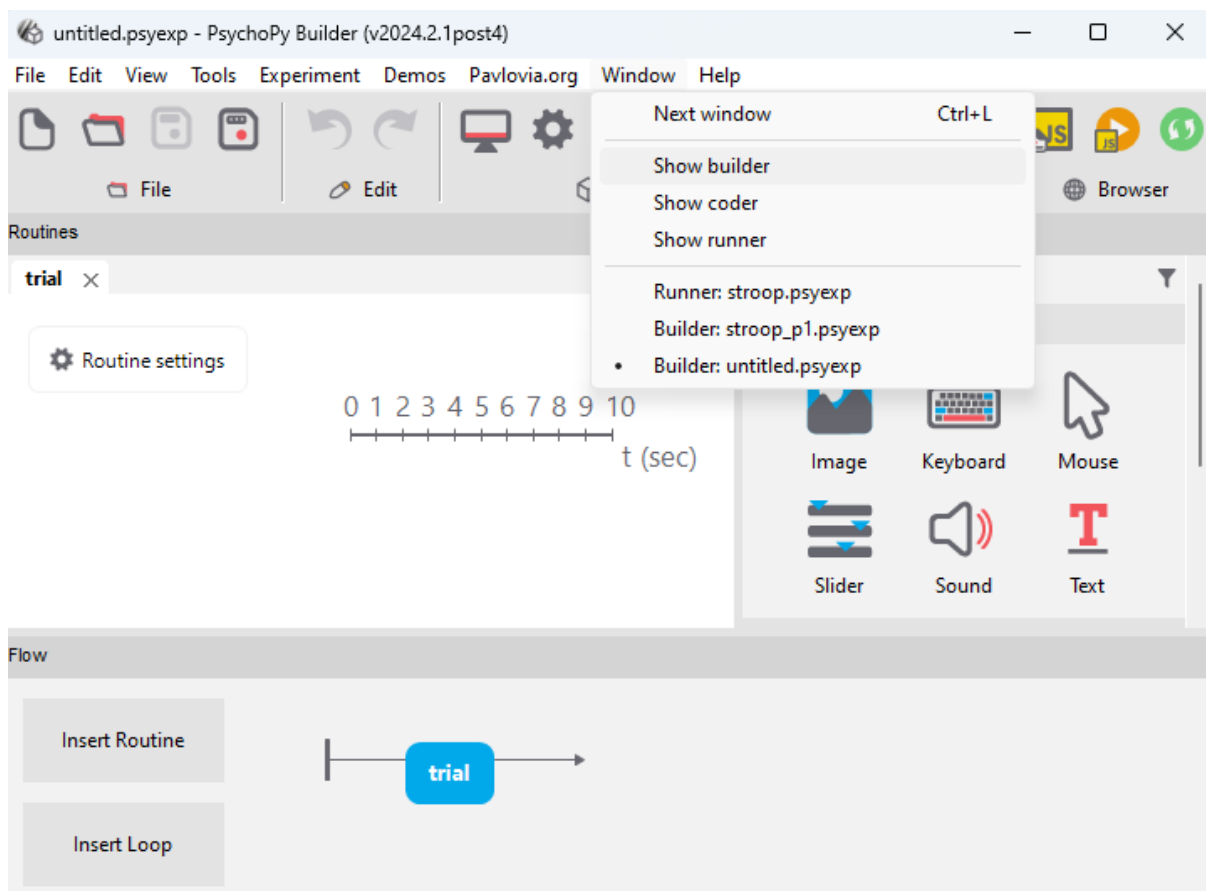
3. PsychoPy: töö komponentidega

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 rutiinielemendile *trial* tekstikomponent ja muuda selle järgmisi prameetreid:

- anna sellele uus nimetus (nt `stroopText`);
- määra elemendi algushetkeks 0.5 ja kestuseks ∞ (lõpmatus ehk kast jääb tühjaks);
- määra elemendi värvuseks punane (*red*);
- määra teksti sisuks „punane“.

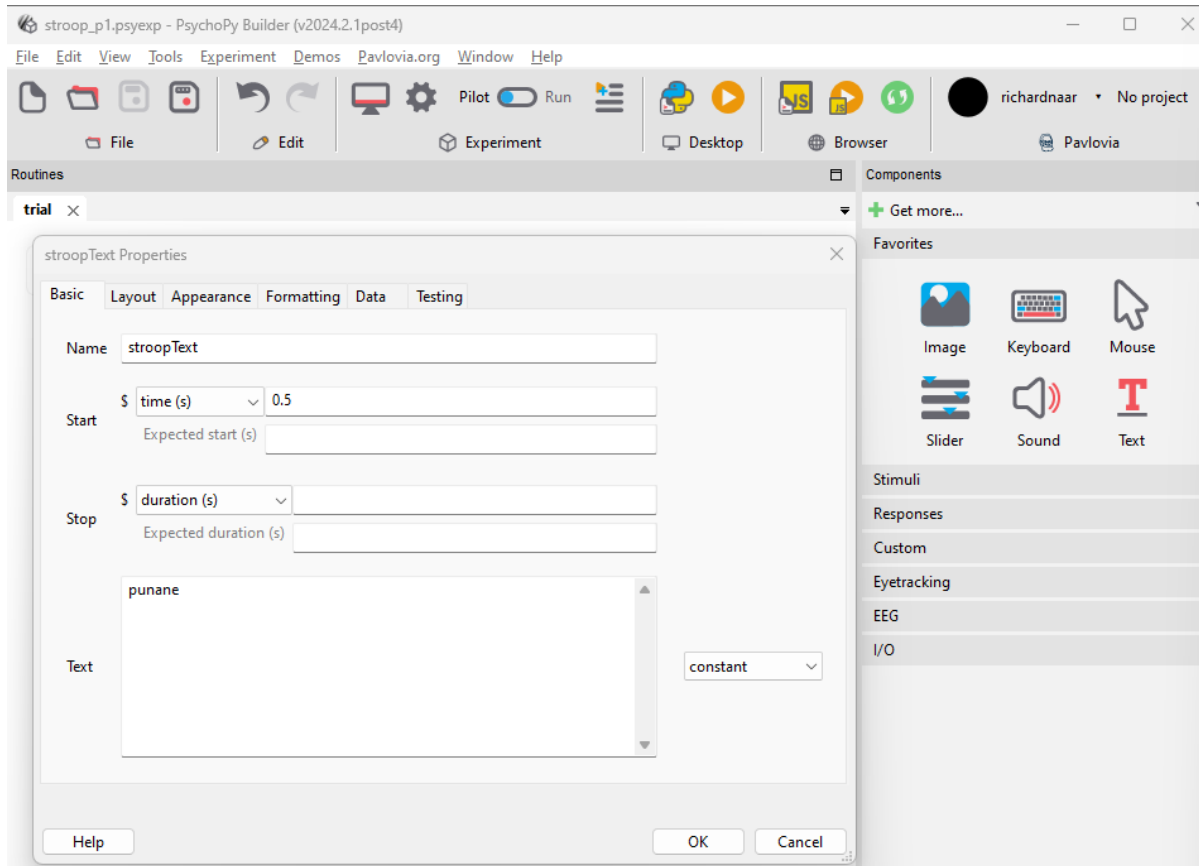
Ülesande lahendamiseks läheme PsychoPy graafilise liidese aknasse.



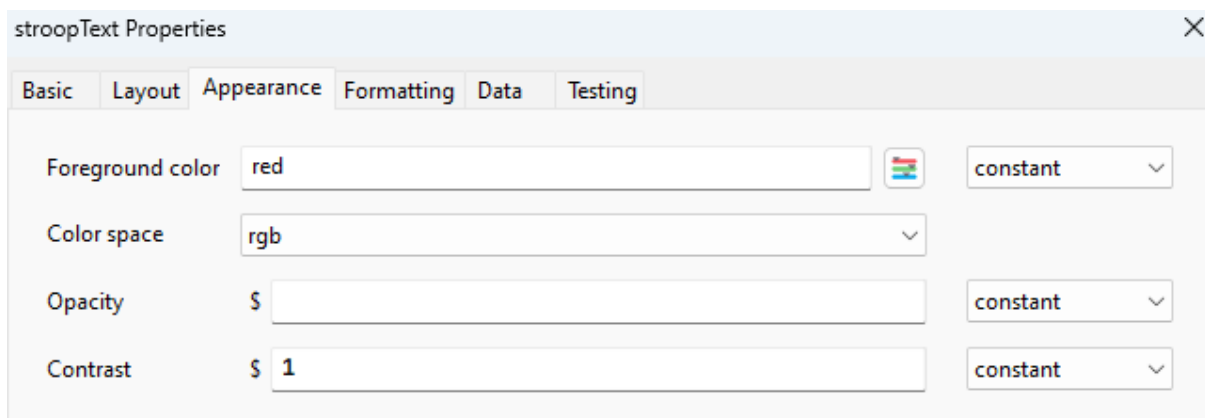
Veendume, et alguses on meil ees tühi eksperiment. Seda on lihtne kontrollida, sest tühjas eksperimendi failis pole veel komponente ja vaikumisi on seal vaid üks rutiinielement (*trial*). Salvestame katsefaili tingimuste failiga samasse kausta (...PsychoPy_experimendid/minu Stroop; salvestamise aken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

Tekstielemendi lisamiseks valime komponentide menüüst: [Components > Stimuli > Text](#) (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 teeme üleval olevate sakkide hulgas **Basic** saki asemel aktiivseks saki **Appearance**. Teksti värvuse punaseks muutmiseks kirjutame **Foreground Color** lahtrisse *red*. Tekstielemendi sisu muutmiseks lähme tagasi **Basic** saki alla ja kirjutame lahtrisse **Text** „punane“.



Tekstikomponendi lihtsätete seadistamine



Tekstikomponendi värvuse seadistamine

Kui muudatused on tehtud, siis vajuta „OK“. Seejärel proovi programmi käivitada (programmi käivitamiseks tee hiireklõps esimesele tumekollasele valge siluetiga ikoonil või

kasuta klaviatuuri otseteed **Ctrl + Shift + R** (Mac: **Cmd + Shift + R**). Programmi käivitamisel ilmub infokastike. Vaikimisi küsitakse seal katseisiku koodi (PsychoPy uuemas versioonis leiab programm vaikimisi sinna ühe arvu, mis jääb vahemikku 0-999999; nii suurt vahemikku on tarvis selleks, et kahele katseisikule juhuslikult sama arvu ei loositaks) ja katsekorra või sessiooni numbrit. Antud hetkel ei oma need suurt tähendust ja võib tühjaks jätta, sest tahame ainult kontrollida, kas programm töötab nii nagu vaja. Programmist väljumiseks vajuta klaviatuuril **Esc** nuppu.

Praktilisi tähelepanekuid

- Eksperiment ei pea valmima ühe korraga – kasulik on pärast iga suuremat muudatust eksperiment käivitada ja kontrollida, kas kõik ikka töötab plaanitul. 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 (eksperimenti sätteid 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.
 - Komponentide sätete menüü avamiseks ja muutmiseks tee komponendil hiireklõps. Komponentide eemaldamiseks vii kursor komponendi peale ja vajuta paremat hiireklahvi ning tekkinud menüüst vali *remove*.
-

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 kursor 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 andmise komponent ja muuda selle järgmisi seadeid:

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

Vastuse registreerimise komponendi lisamiseks valime komponentide menüüst (**Components**): **Responses > Keyboard** (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 vastuse registreerimist alustama. Kui soovid, et vastamise aeg poleks piiratud, siis tuleks **Stop [duration (s)]** lahtr 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).

Vastusekomponendi seadistamine

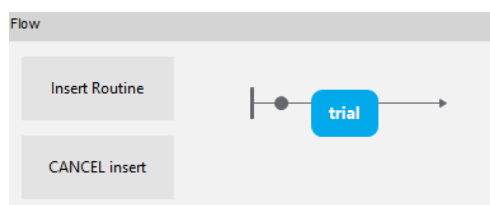
NB!

- 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 lõpu aega 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 tsüklil (*loop*) ja sea rutiinielemendi korduste arvaks 3.

Tsüklil (*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. Algas- 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.



Rutiinielemendile tsükli lisamine

Ülesanne 7. Lisa tingimuste fail esituse tsüklile.

Tsükli lisamisel avaneb menüü, milles saadki tingimuste faili lisada. Kui tsüklil on juba lisatud, siis pääsed tsükli parameetritele ligi sellel hiirega klõpsates. Avanenud menüüs klõpsa **Conditions** lahtri järel paikneval kaustal. 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 .csv/.xlsx formaadis 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).

Rutiinielemendi kordamisega seotud sätted

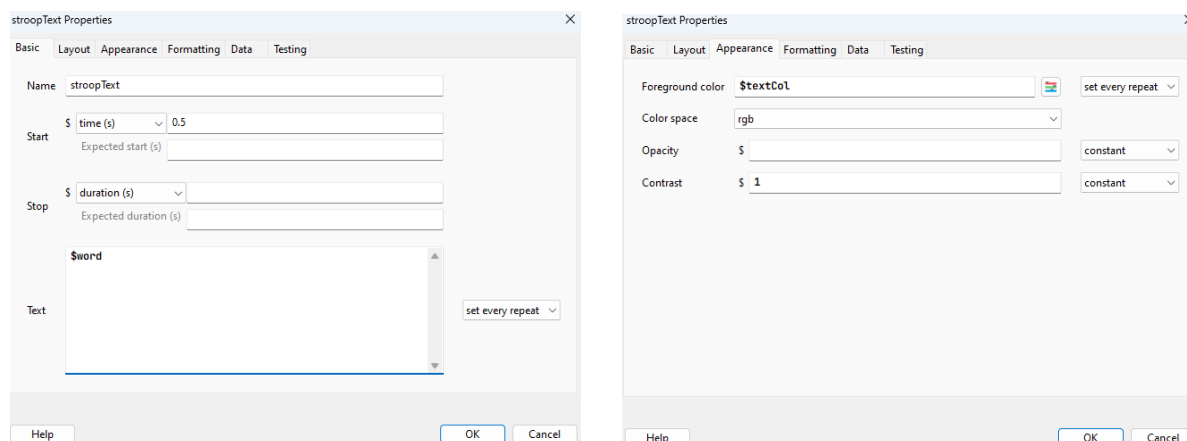
Kui enne tähistas `nReps` \$ lahtrisse sisestatud arv üksikute esituste arvu, siis pärast tingimuste faili lisamist tähistab see seda, mitu korda tingimuste faili katses korratakse. Kui jätame arvu muutmata, tähendab see, et meie katses on kokku 12×3 ehk 36 üksikesitust. Muidugi tahame ka, et programm 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 kuni kõik read on läbi tehtud. Nüüd ajab programm read uuesti segamini ja kordab tehtut kokku `nReps` \$ lahtris määratud korda (antud juhul 3). `loopType` järel oleva valiku peale klõpsates näed, et valikuid on teisi. Sellest, mida teised valikud tähendavad 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 `Appearance` saki all valiku `Foreground Color` järele 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. Muuta tuleks ka sisestuskasti järel olevat muutujat. Vaikimis on seal *constant*, mis tähendab, et tekstikasti sisestatud väärtus katse jooksul ei muutu ehk jääb konstantseks. 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 uuendada².

Lisa nüüd **Basic** saki all ka valiku **Text** lahtrisse veeru nimetus, kuhu eelnevalt eestikeelsed värvinimetused salvestasime (ka siin tuleks PsychoPy-d teavitada, et seda teksti tuleks koodijupina interpreteerida). Valime siingi tekstikasti järel olevate valikute seast *set every repeat*.



Tekstikomponendi seadete muutmine

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

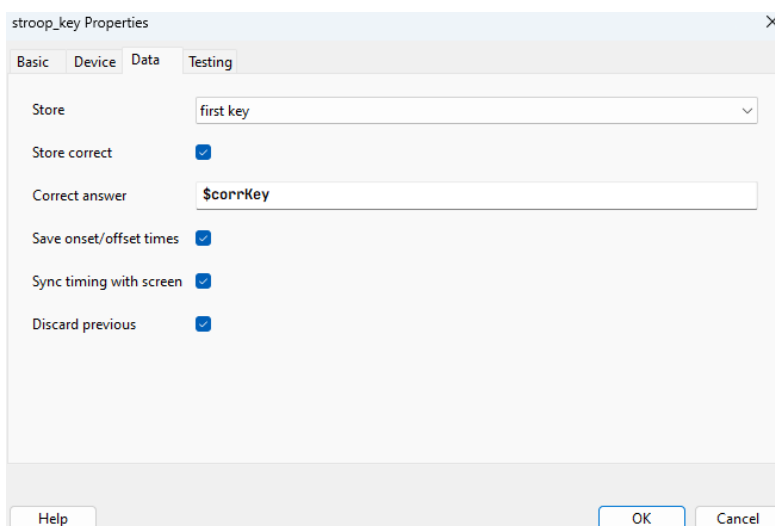
- Tegelikult toimivad *set every repeat* ja *set every frame* valikud käesoleval juhul näiliselt väga sarnaselt, sest PsychoPy võtab iga esituse alguses meie poolt lisatud tingimuste faili vastavatest veergudest ühe väärtuse ning seda väärtust (nt „red“) sama esituse jooksul ei muudeta. Siiski pole päris korrektne järeldada, et polegi tähtis kumba 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 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 kirjuta **Data** saki all olevasse **Correct answer** lahtrisse selle tingimuste

² Kui sa pole kindel, mida siin „ekraani värskenduse“ all silmas peetakse, siis pole midagi. Sellest tuleb põhjalikumalt juttu veebikatsete 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.

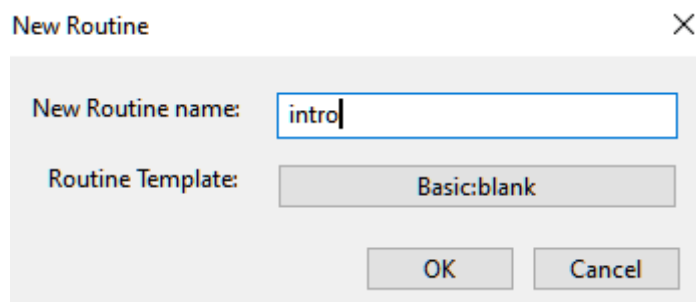
faili veeru nimi, kuhu kirjutasime klahvide inglisekeelsed nimetused. Selleks, et lahter aktiivseks läheks, tee valiku **Store correct** järel linnuke. Sarnaselt tekstikomponendiga tuleks ka siin nime ette dollari märk (\$) lisada.



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.

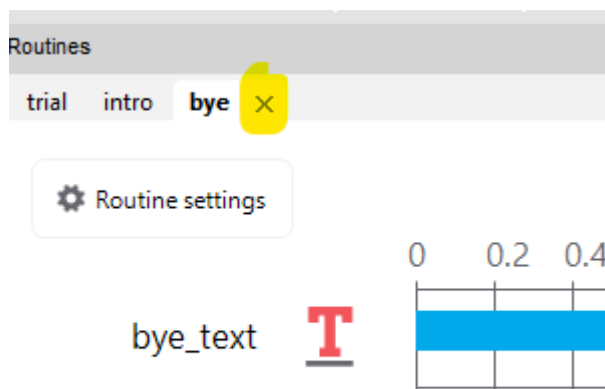


Rutiinielemendi nimetamine

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 ajajoonele must täpp, mis lubab rutiinikomponendile kuni järgmise hiireklõpsuni kohta otsida.

Rutiinielementide nimetamine ja eemaldamine

- Rutiinielemendi asukoha vahetamiseks tuleks menüüst sama rutiin uuesti valida ja soovitud kohta asetada. Soovimatul kohal paikneva rutiinielemendi eemaldamiseks tee rutiinikomponendil parem klõps ja vali *remove*. See valik ei kustuta rutiinielementi vaid peidab selle. Kui soovid rutiinielemendi kustutada, siis tuleks rutiin menüüriba all paiknevast tabi ristist sulgeda (vt alumine joonis).
- 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.



Rutiinielemendi kustutamine

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 vajutades ta tagasi tuua (vasakule poole näitav hall nool); 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 rutiinieleменти 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 ilmutat infokasti selliselt, et sinna saaks sisestada katseisiku numbrit, soo ja vanuse.

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 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.

Field	Default		
participant	f"{randint(0, 999999):06.0f}"	+	-
vanus		+	-
sugu	['Naine', 'Mees', 'Muu']	+	-

Buttons: Help, OK, Cancel

Infokasti seadete muutmine

Infokastikest on hea kasutada ka katse parameetrite muutmiseks, sest sinna sisestatud infot saad programmi sees kasutada. Näiteks võid teha lahtri „esituste arv“, mis hakkaks määrama seda, mitu seeriat iga tingimustabeli rea kohta esitatakse jms. Seda, kuidas infokastikesse sisestatud väärtusi programmi sees vaadata vaatame juba järgmises praktikumis. Kui see info kiirelt kokku võtta, siis infokastikese sisu nägemiseks tuleks kirjutada Pythoni süntaks, mis võtaks infokastikese struktuurist meid huvitava infokillu. Näiteks katseisiku numbri leidmiseks võiksime kasutada järgmist süntaksit: `expInfo['Participant']`.

NB! Participant veergu kasutatakse andmefaili nimetamisel ja seetõttu annab programm selle veeru muutmisel veateate. Kui seda veergu muuta, siis tuleks vastav muudatus teha ka faili nimetamise lahtris, mis asub eksperimendi sätete **Data** saki all.

stroop_p1 Properties

Basic Screen Audio Online Eyetracking Input Data

Data filename: `u'data/%s_%s_%s' % (expInfo['participant'], expName, expInfo['date'])`

Data file delimiter: auto

Teine moodus infokastikese vaikeväärtuste salvestamiseks on teha nendest loend, mille hulgast kasutaja saaks endale sobiva valida. Selleks tuleks vaikeväärtused Pythoni sõnaraamatu (*dictionary*) kujul **Default** lahtrisse sisestada. See ei tähenda tegelikult midagi muud, kui seda, et meie sisestatud vaikeväärtuste loend peaks paiknema kandiliste sulgude vahel ja iga element peaks olema jutumärkides ja teistest elementidest komaga eraldatud. Soo puhul näiteks nii: `['N', 'M']`.

stroop_p1 Properties

Basic Screen Audio Online Eyetracking Input Data

Experiment name: stroop_p1

Run mode: Piloting

Use PsychoPy version: [dropdown]

Enable escape key: ☒

Enable 'rush' mode: ☐

Show info dialog: ☒

Field	Default
participant	f'"{randint(0, 999999):06.0f}"
gender	['N', 'M']
age	

stroop_p1

participant: 950178

gender: N

age: [empty field]

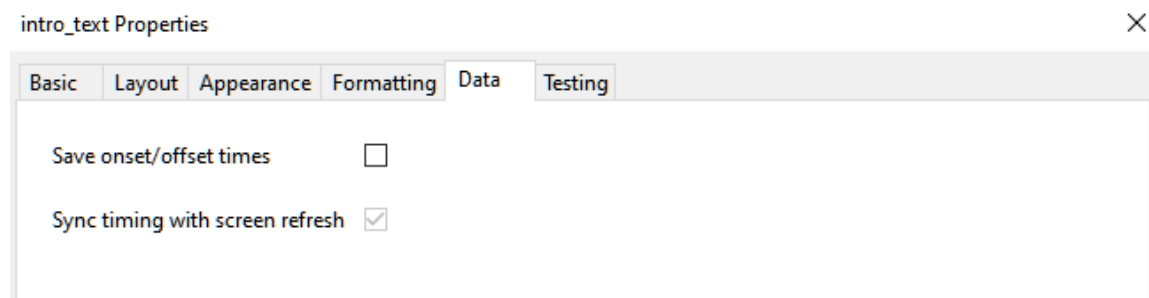
OK Cancel

Infokastikese lahtri rippmenüüna seadistamiseks tuleks vaikeväärtused jutumärkides ja komaga eraldatult kandiliste sulgude vahele lisada

Ülesanne 12. Muuda *trial* tsüklit selliselt, et katse esitataks iga tingimuste tabeli 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 andmed kokku 12 esituse kohta. Lisaks nendele veergudele, mille defineerisime tingimuste tabelis ja infokastikeses, on andmetabelis veel teisigi veerge. Andmetabelis on näiteks veerud, mis viitavad sellele, millal 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 või väljalülitamise kohta infot andmefaili ei salvestata.



Komponenti sisselülitamisega seotud aja salvestamise väljalülitamine

Treeningseeriad

Enamikel juhtudel soovime esitada katseisikule ka treeningseeriaid. Treeningseeriade esitamiseks pole tüüpiliselt tarvis muud, kui olemasoleva tsükli (*loop*) ümber teha uus tsükkel, mis loeb sisse tingimuste faili, mis defineerib, mitu korda tingimuste tabelit treeningseerialtel ja katseseerialtel esitatakse.

Samm-sammult liikudes tuleks kõigepealt luua uus tingimuste tabel, kuhu on kantud info treeningseeriade ja eksperimenditaoliste seeriade arvu kohta. Lisaks võime defineerida siin, millist teksti peaks katseisikule enne vastavaid seeriaid esitama. Võime selle salvestada näiteks `block_info.xlsx` failina.

	A	B	C	D	E
1	block	repTimes	block_intro		
2	training		1 Nüüd algavad treeningseeriad. Jätkamiseks vajuta palun tühikut...		
3	experiment		3 Nüüd algab katse. Jätkamiseks vajuta palun tühikut...		
4					

Seejärel tuleks katseskeemi menüüs lisada olemasoleva „trials“ aasa ümber uus tsükkel ja seadistada see selliselt, et tingimuste tabeli ridu loetaks järjestikuliselt (Loop type: sequential). Lisaks võiks **is trials** eest võtta ära linnukese, sest tegemist pole mitte üksikute esitustega vaid plokkidega. Esituste arvuks peaks olema 1, sest tahame iga `block_info` tabeli rida esitada ühe korra. Enne tsükli lisamist tuleks `block_info` tabel tsükli koosseisu lülitada.

blocks Properties

Name



Loop type

Is trials ☐

Num. repeats \$

Selected rows

Random seed \$

Conditions  

2 conditions, with 3 parameters
[block, repTimes, block_intro]

Help OK Cancel

Kuna esituste arv sõltub sellest, kas tegemist on treeningseeriatega või eksperimentaalsete seeriatega, siis tuleks muuta ka trials aasa sisu ja kirjutada **Num. repetas** lahtrisse veeru nimi, kuhu sisestasime, mitu korda tuleks tingimuste tabeli ridu korrata (repNum).

trials Properties

Name



Loop type

Is trials ☒

Num. repeats \$

Selected rows

Random seed \$

Conditions  

12 conditions, with 4 parameters
[textCol, word, corrKey, congruency]

Help OK Cancel

Nüüd on jäänud veel lisada blocks süntaksi sisse (ja trials süntaksist välja) rutiinielement, mis esitaks katseisikutele vastava plokiga sobituva teksti, mille salvestasime block_info tabeli

block_intro veergu. See rutiinielement töötab sarnaselt eelnevalt defineeritud intro rutiinile, kuid ainult selle vahega, et teksti sisu loetakse block_info tabeli veerust block_intro.

warning_text Properties

Basic Layout Appearance Formatting Data Testing

Name **warning_text**

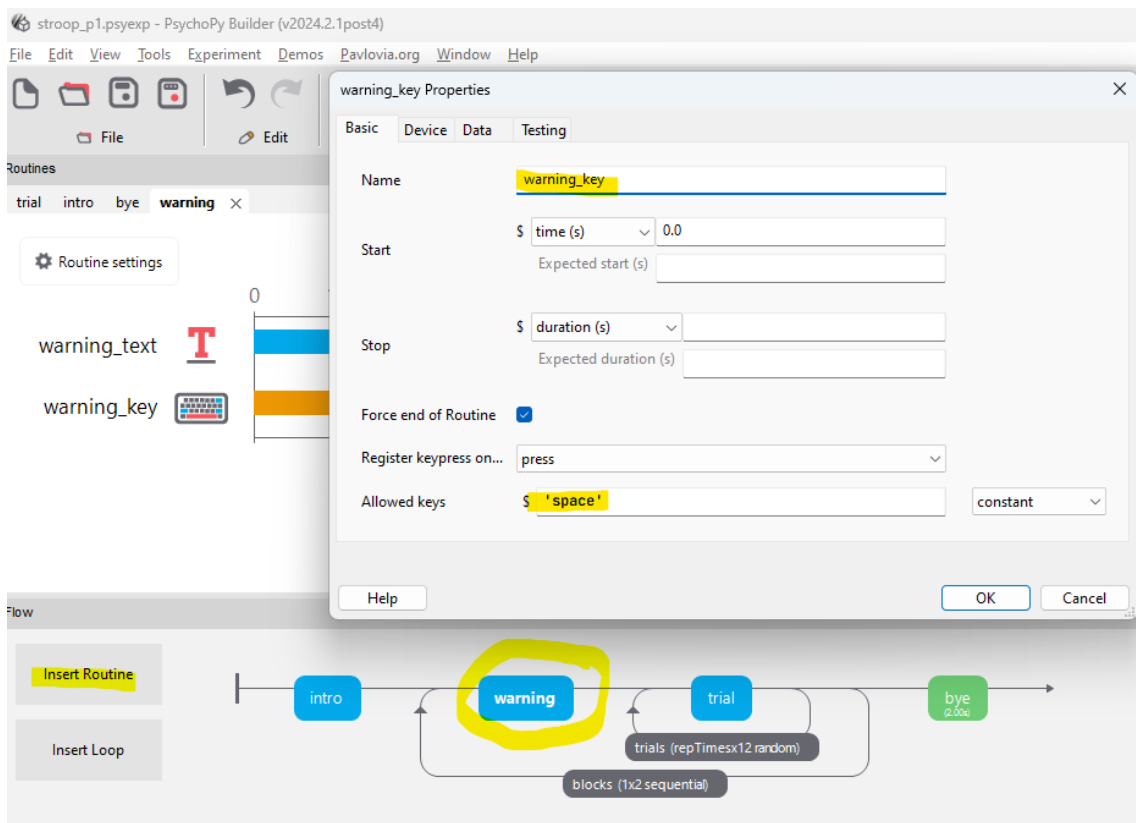
Start \$ time (s) 0.0
Expected start (s)

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

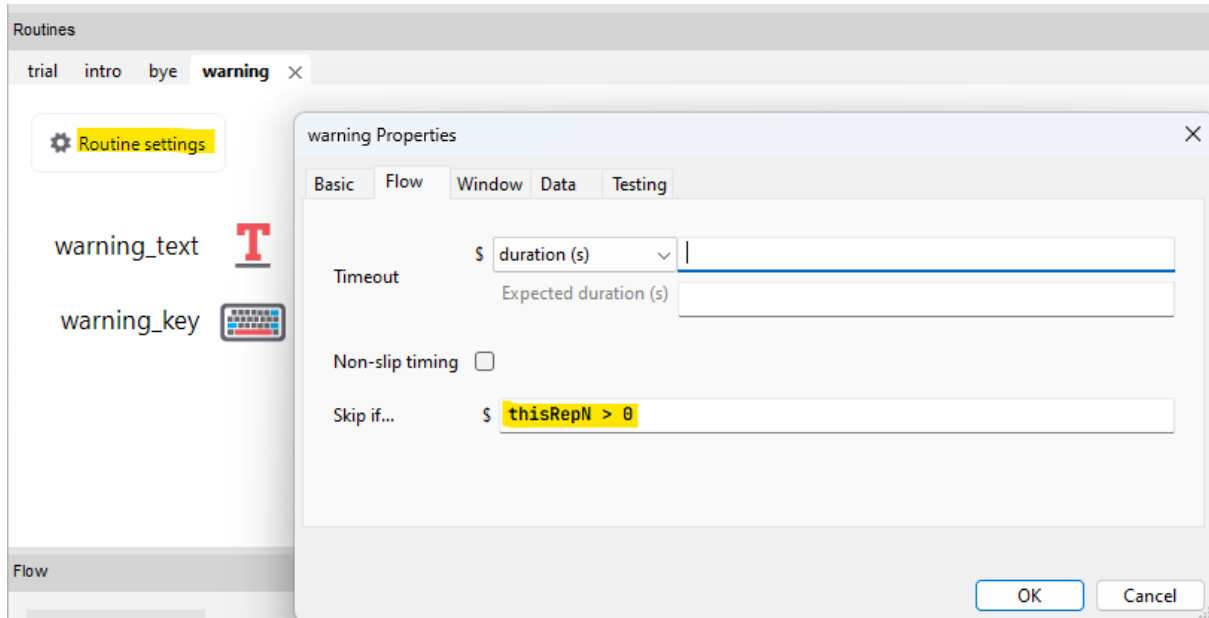
Text **\$block_intro**

set every repeat

Help OK Cancel



Kui nüüd seadistame „warning“ rutiini selliselt, et rutiin käivitatakse ainult juhul, kui tegemist on esimese kordusega kummaski plokis, siis näidatakse hoiatusteksti ainult treeningseeriade ja katseseeriade alguses ehk täpselt nii nagu vaja (**Routine settings** > **Flow** ja seejärel **Skip if...** lahtrisse sisestame: $\text{thisRepN} > 0$). Selle üle, mitmenda kordusega on tegemist peab järke muutuja thisRepN .



Kohtame sama muutujat ka andmetabelis. Kõiki andmetabelis olevaid muutujaid saab katse juhtfailis analoogselt kasutada.

Lisaülesanded

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

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