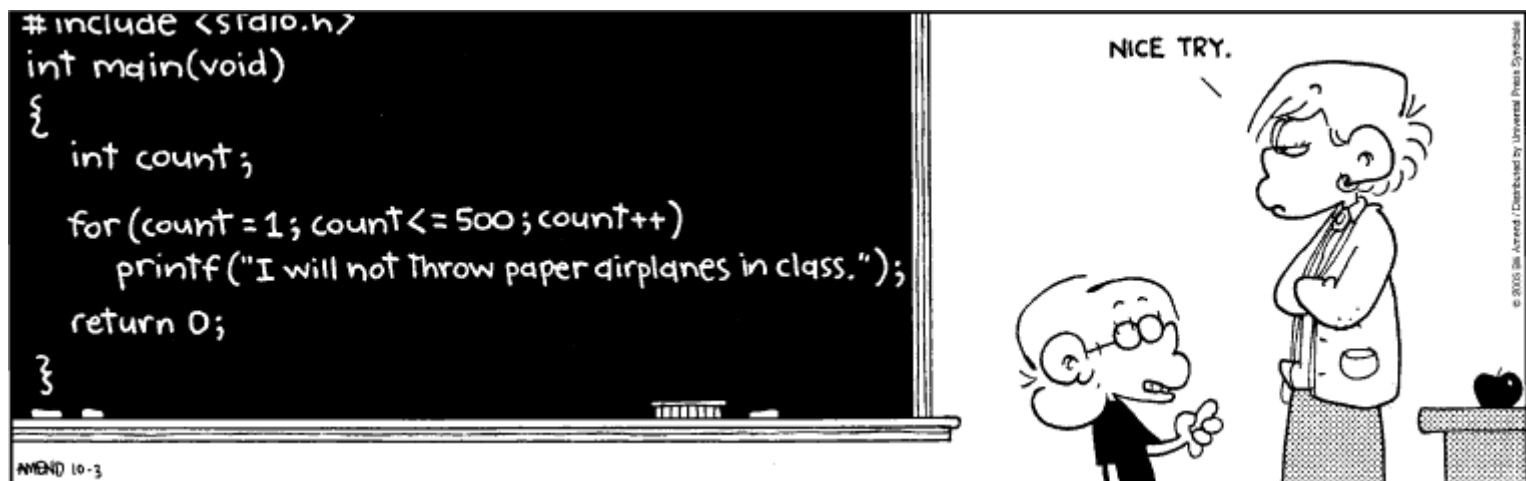


Módszertani és informatikai alapismeretek (Alkalmazott lélektan speciális kollégium)

Soltész-Várhelyi Klára

OpenSesame

Algoritmizálás (leg)alapjai



Algoritmizálás (leg)alapjai

Codie ötlete alapján



<http://www.getcodie.com>

Ki Codie?

- Ki Codie?
 - Codie egy robot, akinek a viselkedése okostelefonnal szabályozható (programozható)
 - Magyar fejlesztésű projekt (www.getcodie.com)
 - Segítségével a gyerekek akár 5 éves kortól elsajátíthatják a programozáshoz szükséges logikus gondolkodás alapjait.
- Mit tud Codie csinálni?
 - Codie ma már kicsit máshogy (okosabban) viselkedik, de mi az órán ezeket a funkciókat fogjuk használni:
 - Viselkedés:
 - Előre gurul 10cm-t
 - Fordul adott fokot
 - Megvizsgálja, hogy van-e előtte 10cm-en belül akadály
 - Várakozik adott időt
 - Programozás:
 - Menj előre 10cm-t 
 - Vizsgáld meg, van-e előtted valami 
 - Csináld többször 

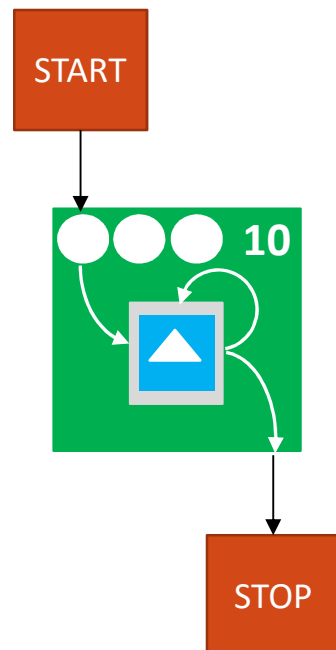
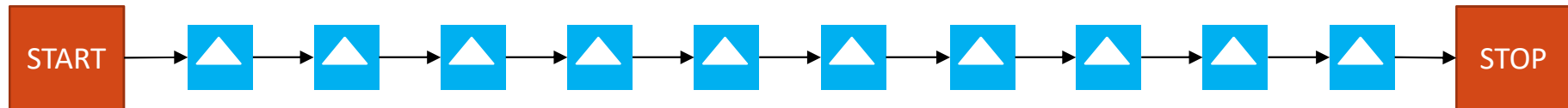
Fordulj



Várakozz



Menj előre 1 métert!



Ezt nevezzük **for ciklus**nak

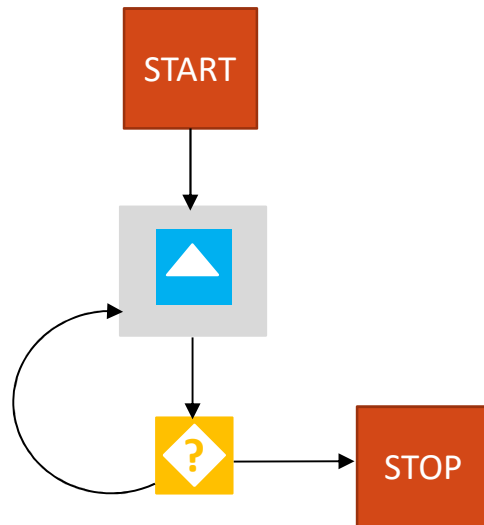
Ez a 3 lehetséges **loop** (hurok) közül az első

A belső parancsok sorozatát nevezzük **blokk**nak

A változót, mely számolja, hányszor futott már le a blokk
nevezzük **futóindex**nek, és általában i-vel vagy j-vel jelöljük

A for ciklus i alkalommal hajtja végre a blokk utasításait

Menj el a falig!



Ezt nevezik **do-while** ciklusnak

A 3 lehetséges **loop** közül a második

A parancsok sorozatát nevezzük **blokk**nak

A feltétel ellenőrzését **feltétel**nek vagy **if kondíció**nak nevezzük

A do-while végrehajt egy blokkot, majd ellenőrzi, hogy egy adott feltétel fennáll-e még, és addig ismétli a blokkot, amíg a feltétel hamisra nem vált.

De ne dönts fel a vázát!

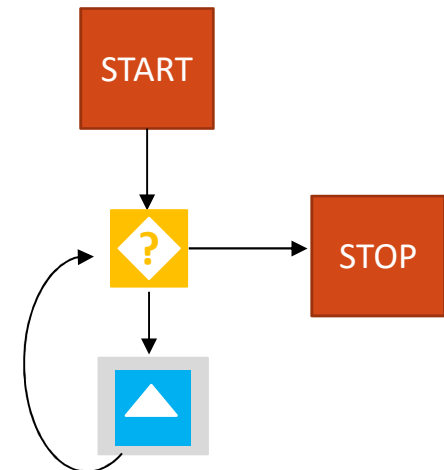
Ezt nevezik **while** vagy **while-do** ciklusnak

A 3 lehetséges **loop** közül a harmadik

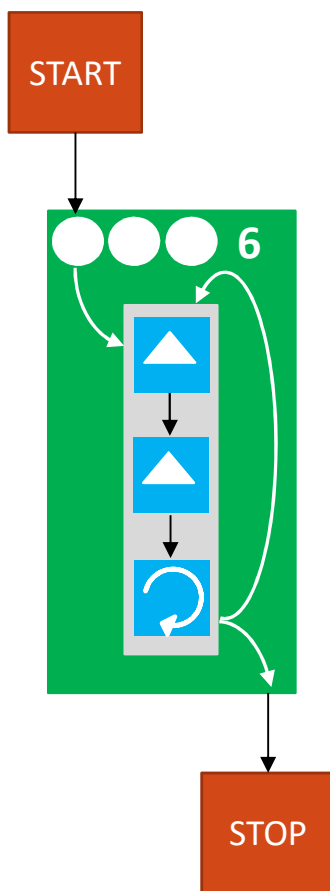
A parancsok sorozatát nevezzük **blokk**nak

A feltétel ellenőrzését **feltétel**nek vagy **if kondíció**nak nevezzük

A while ciklus először ellenőrzi a feltételt, majd ha teljesül, végrehajtja a blokkot, és ezt addig ismétli, amíg a feltétel igaz.



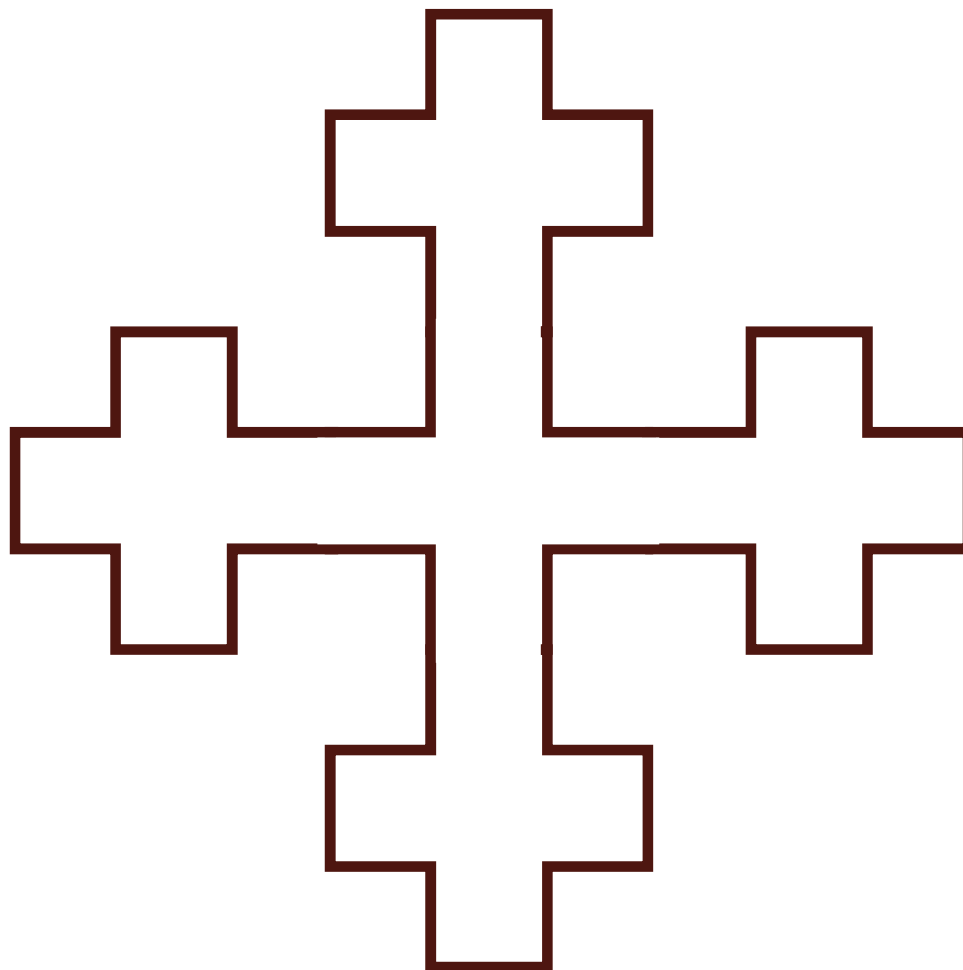
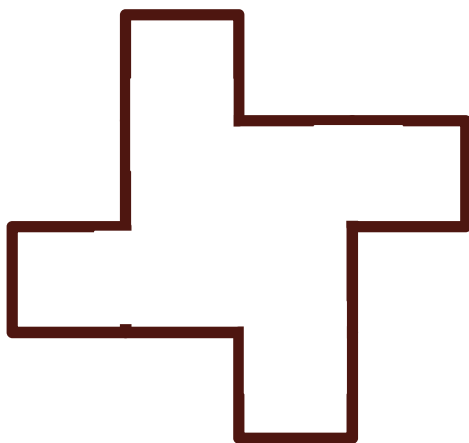
Járj körbe egy 20cm oldalhosszúságú hatszög alakot!



A hatszög bejárásához előre kell gurulni kétszer 10cm-t, majd elfordulni 60 fokot, és ezt hatszor megismételni

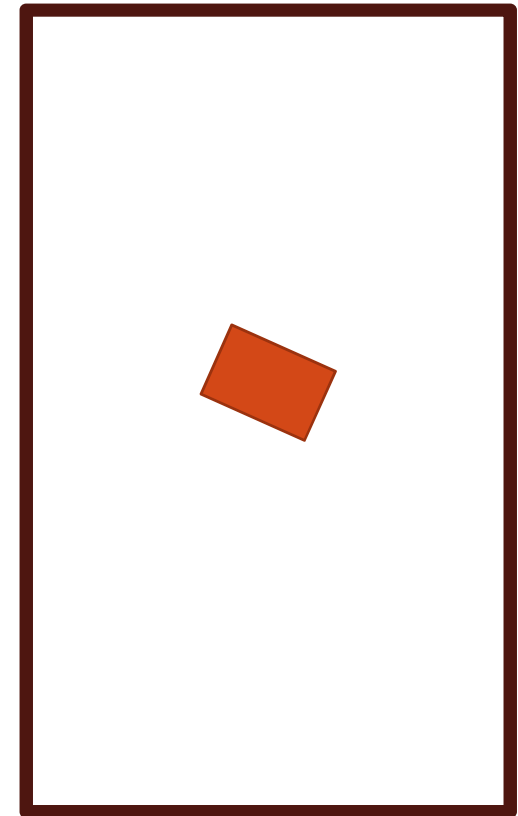
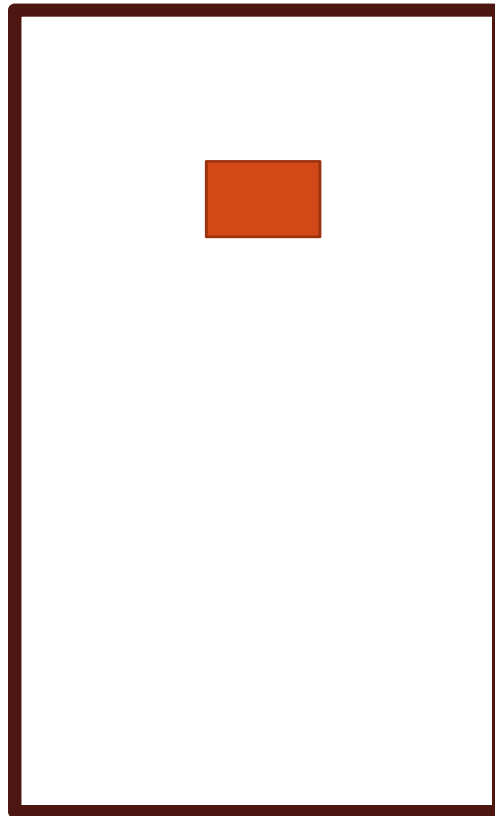
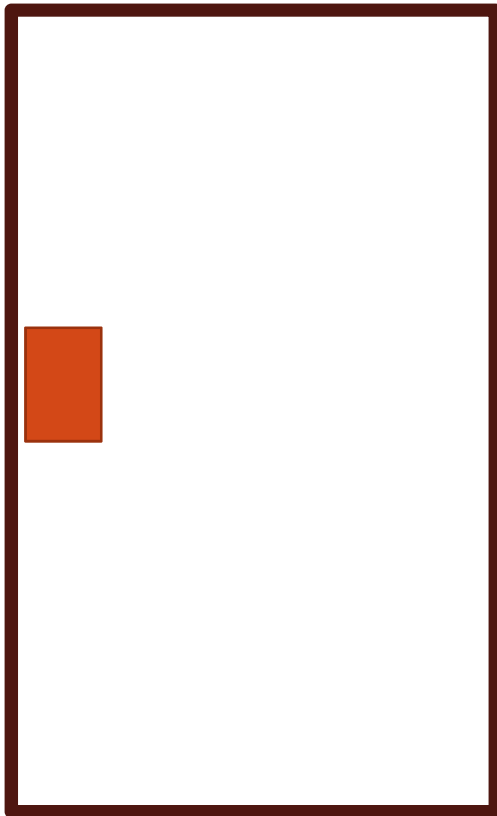
További feladatok:

- Járd körbe a következő alakokat!



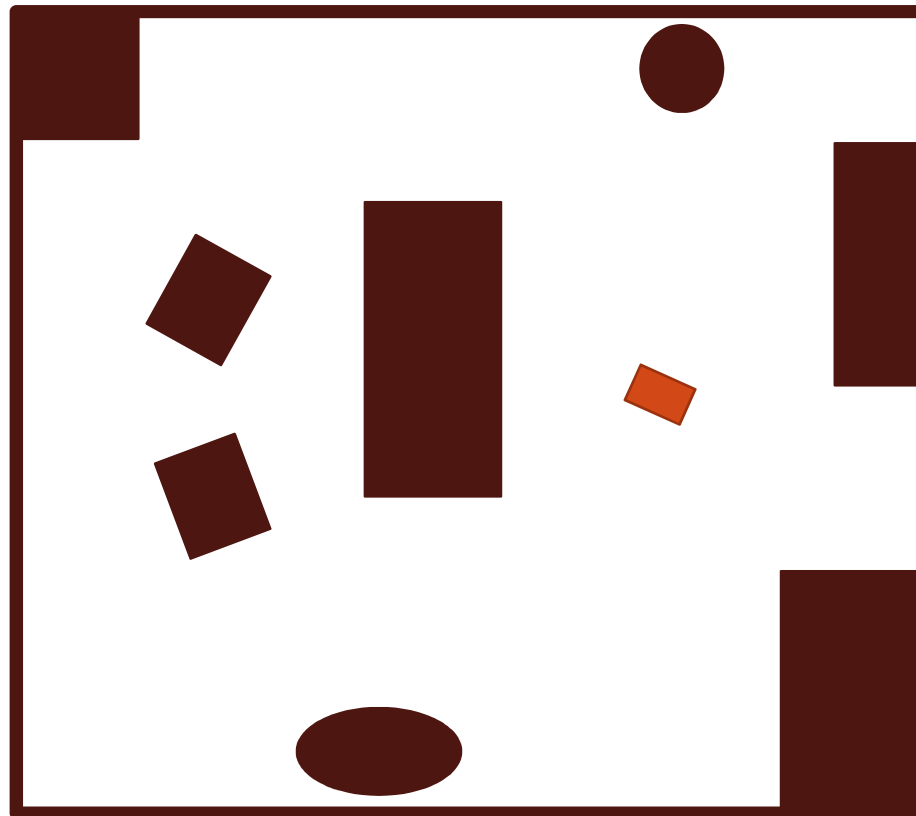
További feladatok:

- Menj körbe a téglalap alapú doboz belső fala mentén a következő kezdőpozíciókból (legalább egyszer haladj el minden oldal mellett)!

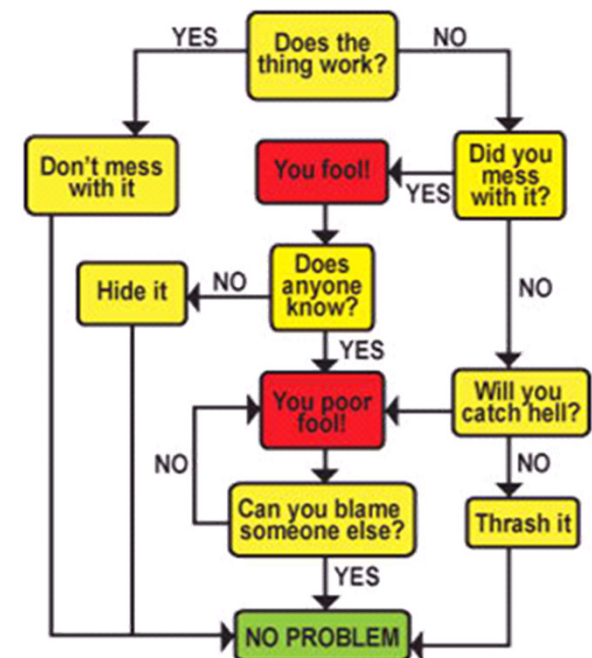


További feladatok:

- Viselkedj porszívó robotként!
- Megy előre amíg tud, ha akadályba ütközik, akkor addig forog, amíg szabad teret nem észlel maga előtt. Nincs automatikus STOP, majd mi leállítjuk, ha elég tiszta a lakás.

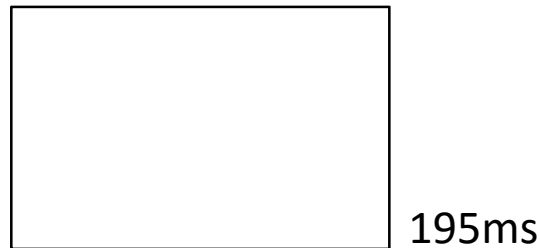


A teszt vázának leírása

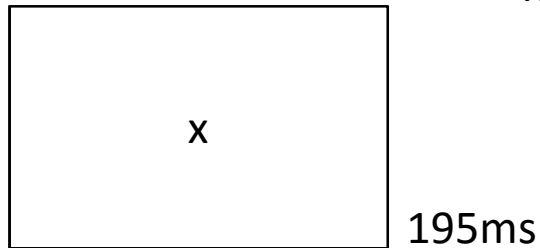


Egyszerű reakcióidő

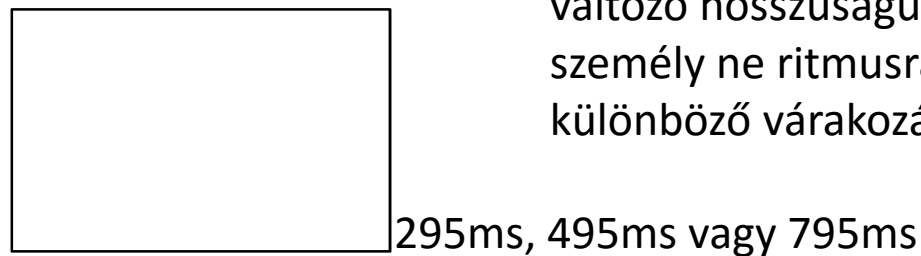
Üres mezővel kezdünk, hogy a fixációs kereszt ne közvetlenül az előző inger után jöjjön.



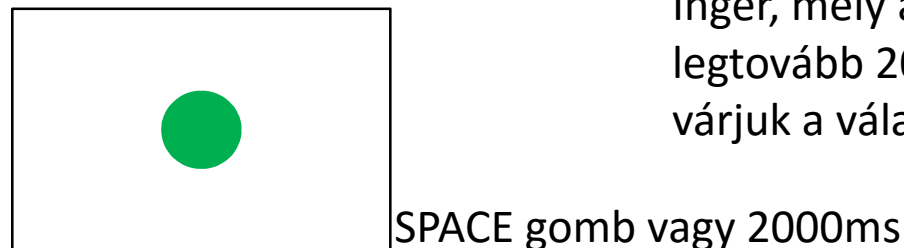
Fixációs kereszt (ebben a kísérletben fix hosszúságú) – jelzi, hogy a kísérleti személy készüljön fel a következő ingerre



A fixációs kereszt után egy üres (vagy semleges) mező jön. Egyszerű reakcióidő vizsgálatban érdemes változó hosszúságúra hagyni, hogy a kísérleti személy ne ritmusra reagáljon. Használjunk pl. 3 különböző várakozási időt: 295, 495, 795 ms-ot!



Inger, mely a gomb lenyomásáig, de legtovább 2000 ms-ig van jelen, eddig várjuk a válasz



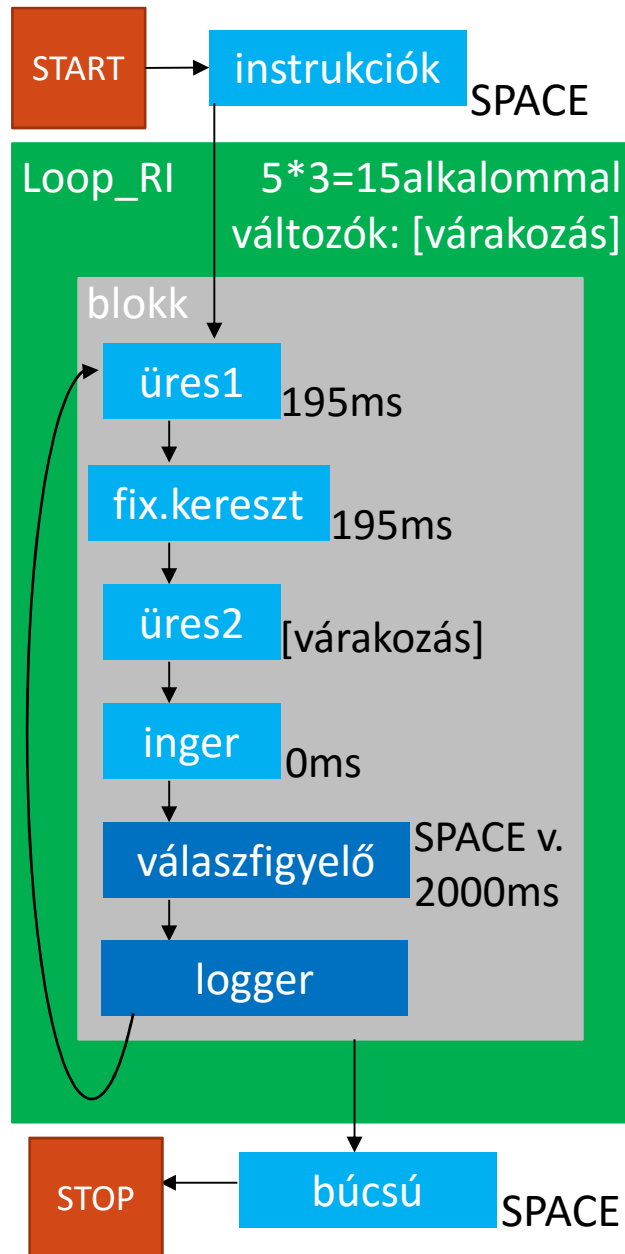
idő

Időzítés

- Ahol fontos a pontos időzítés:
- Képernyőfrissítés
 - A legtöbb monitor frissítési frekvenciája 60Hz: 1 másodpercben (1000ms) 60 váltás, tehát egy kép $1000/60 = 16,6$ ms-ig van fenn (ennél bonyolultabb, mert a frissítés végigfut a képernyőn, de ezzel most nem foglalkozunk)
 - Ez azt jelenti, hogy hiába szeretnénk valamit 5ms-onként megváltoztatni, az nem fog a képernyőn megjelenni.
 - **Ezért ha pontos időzítéseket akarunk, fontos, hogy a késleltetések mindig 16,67 többszörösei legyenek (kerekítve).**
- Előkészítés
 - A másik probléma, hogy a változtatás pillanatára már készen kell lennie az új képnek, erre hagyjunk a programnak néhány ms-ot
 - **Ezért az időzítés legyen mindig 5ms rövidebb, mint amit tényleg szeretnénk.**
- Milyen frissítések használhatók például?

• $6 * 16,67 - 5 \approx 95\text{ms}$	$24 * 16,67 - 5 \approx 395$
• $9 * 16,67 - 5 \approx 145\text{ms}$	$45 * 16,67 - 5 \approx 745$
• $10 * 16,67 - 5 \approx 162\text{ms}$	$60 * 16,67 - 5 \approx 995$

Egyszerű RI



A kísérlet az **instrukciókkal** kezdődik. A ksz. a **SPACE** gomb lenyomásával jelzi, hogy elolvasta az instrukciókat, és mehetünk tovább.

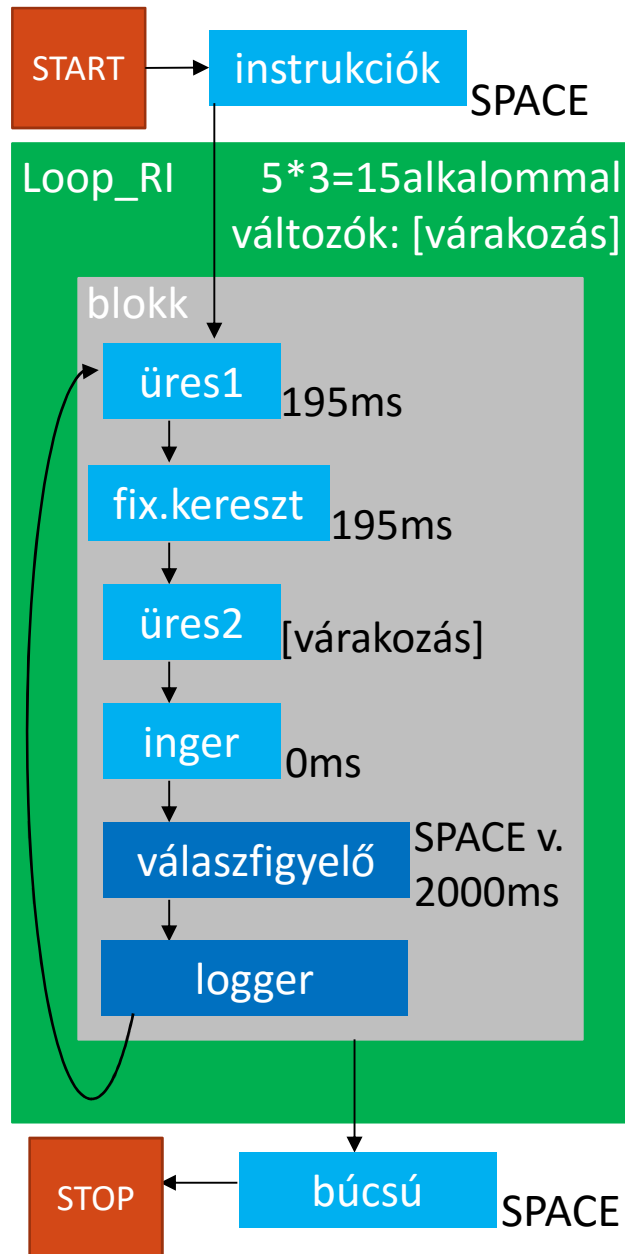
Loop_RI változó tábla

[várakozás]
295
495
795

Ezt követően jön maga reakcióidő kísérlet. Háromfajta késleltetéssel dolgozunk, minden késleltetést 10-szer fogunk használni, így összesen 15 mérés lesz a vizsgálatunkban (élesben 10 helyett 50-100 mérést szoktunk használni). Ezeket összeszervezzük egyetlen **loopba**.

A loop tartalmazza a változó táblát, most egyetlen változónk van, a [várakozás]

Egyszerű RI

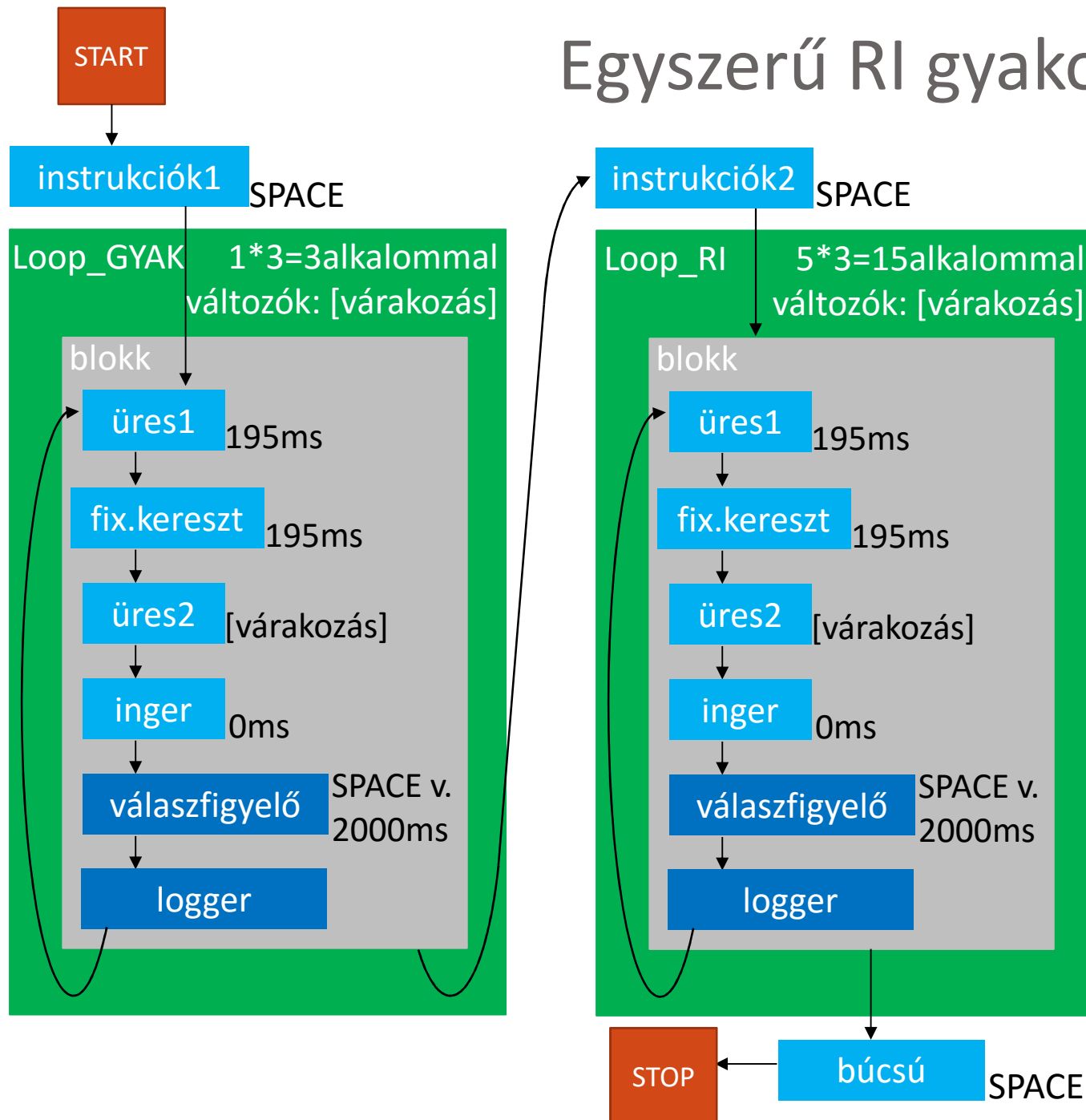


A loopon belül a **blokk**ban a következő események lesznek (ezek ismétlődnek 15-ször):

1. 195ms-ig **üres** a képernyő.
2. 195ms-ig **fixációs kereszt**, a ksz. felkészül az ingerre.
3. Ezt követi egy változó időhosszúságú **üres** képernyő.
4. Majd megjelenik az **inger**, amire a kísérleti személynek reagálni kell.
5. Az inger megjelenése után azonnal (0ms) továbblépünk a következő egységre, ami a **billentyűzet válaszfigyelője**, így tudjuk a ksz. választ befogni. Ha 2sec-ig nem reagál (nem nyomja le a SPACE gombot), akkor tovább lépünk, és ahhoz a méréshez nem fog RI tartozni.
6. Végül még el is kell mentenünk a mért adatokat a **logger**rel.

Ha megvolt a 15 mérés, kilépünk a loopból, **megköszönjük** a ksz.-nek, hogy részt vett a vizsgálatban, és kilépünk.

Egyszerű RI gyakorló fázissal

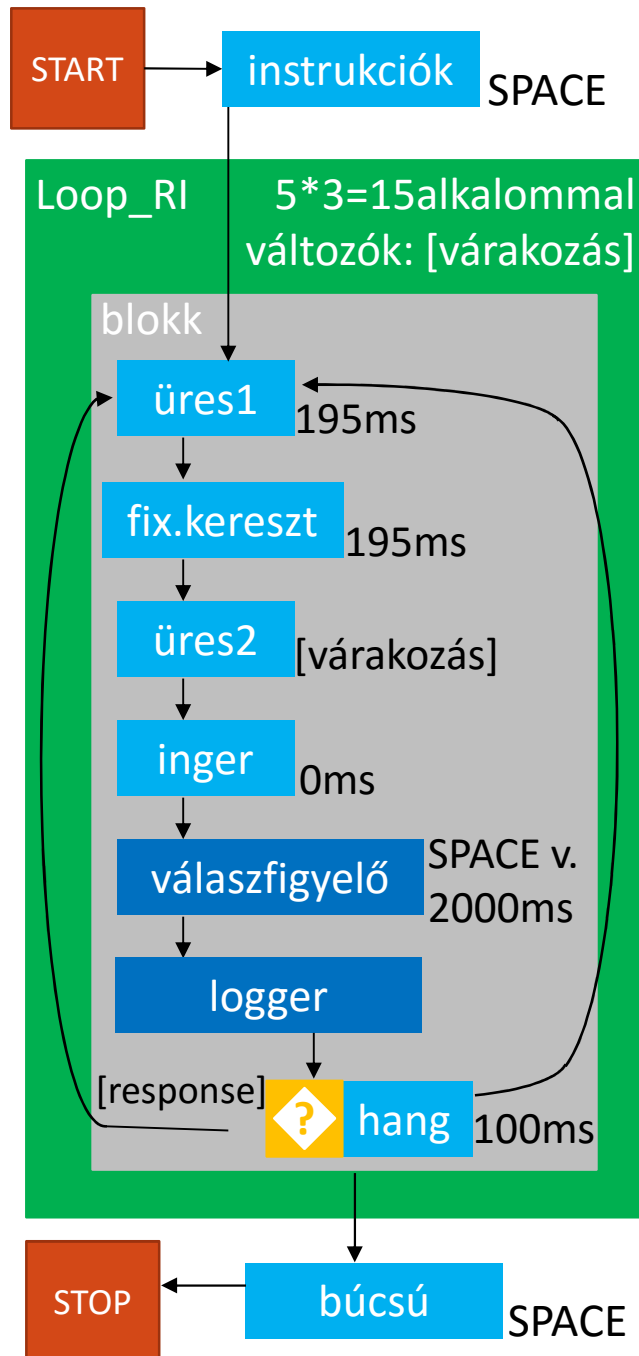


A gyakorlás során kevesebbszer futtatjuk le azt, ami a kísérlet során történni fog.

A blokk mindkét esetben ugyanaz, ezért ezeket összelinkelhetjük, így csak egyszer kell megvalósítani.

Az éles teszt előtt a ksz-nek szólunk, hogy a gyakorlás véget ért, és kezdődik a mérés.

Egyszerű RI hangjelzéssel



Ha a ksz. Nem válaszolt 2 másodpercen belül, akkor lehet, hogy nem figyel, jelezzünk neki hangingerrel.

A hangingeret csak akkor kell leadni, ha nem érkezett válasz az ingerre, tehát a válaszfigyelő által rögzített **[response]** változó értéke **None**.

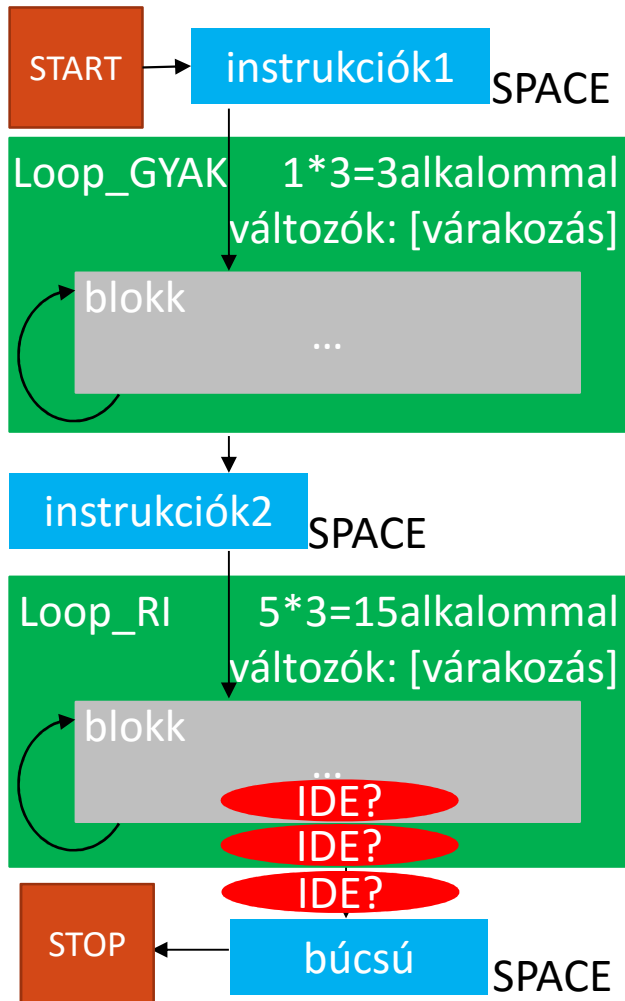
[response] változó: logger által automatikusan generált változó, a kísérleti személy választ tartalmazza: ha lenyomta a **SPACE** billentyűt, az értéke **space**, ha nem nyomta le, az értéke **None**.

Egy **if** kondícióval megvizsgáljuk, hogy a **[response]** változó értéke **None**-e. Ha igen, adjuk ki a hangjelzést. Ha nem, akkor nem kell hangjelzés, mehet tovább a loop.

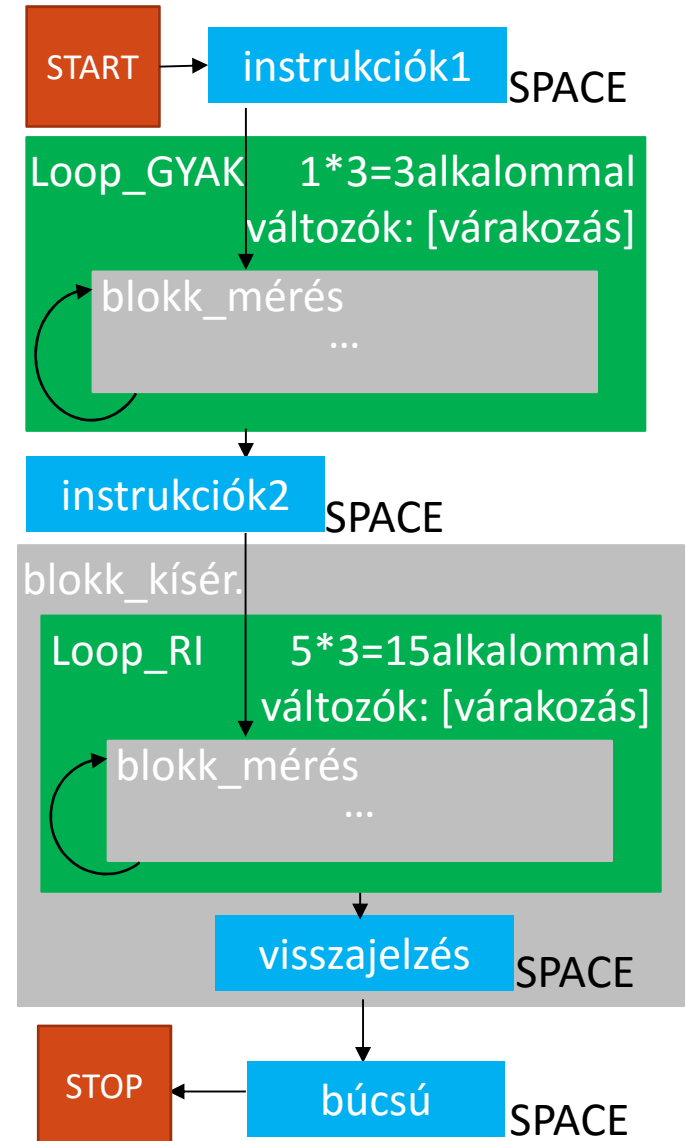
Egyszerű RI visszajelzéssel

Szeretnénk visszajelzést adni a ksz-nek a teljesítményéről. Hova tegyük?

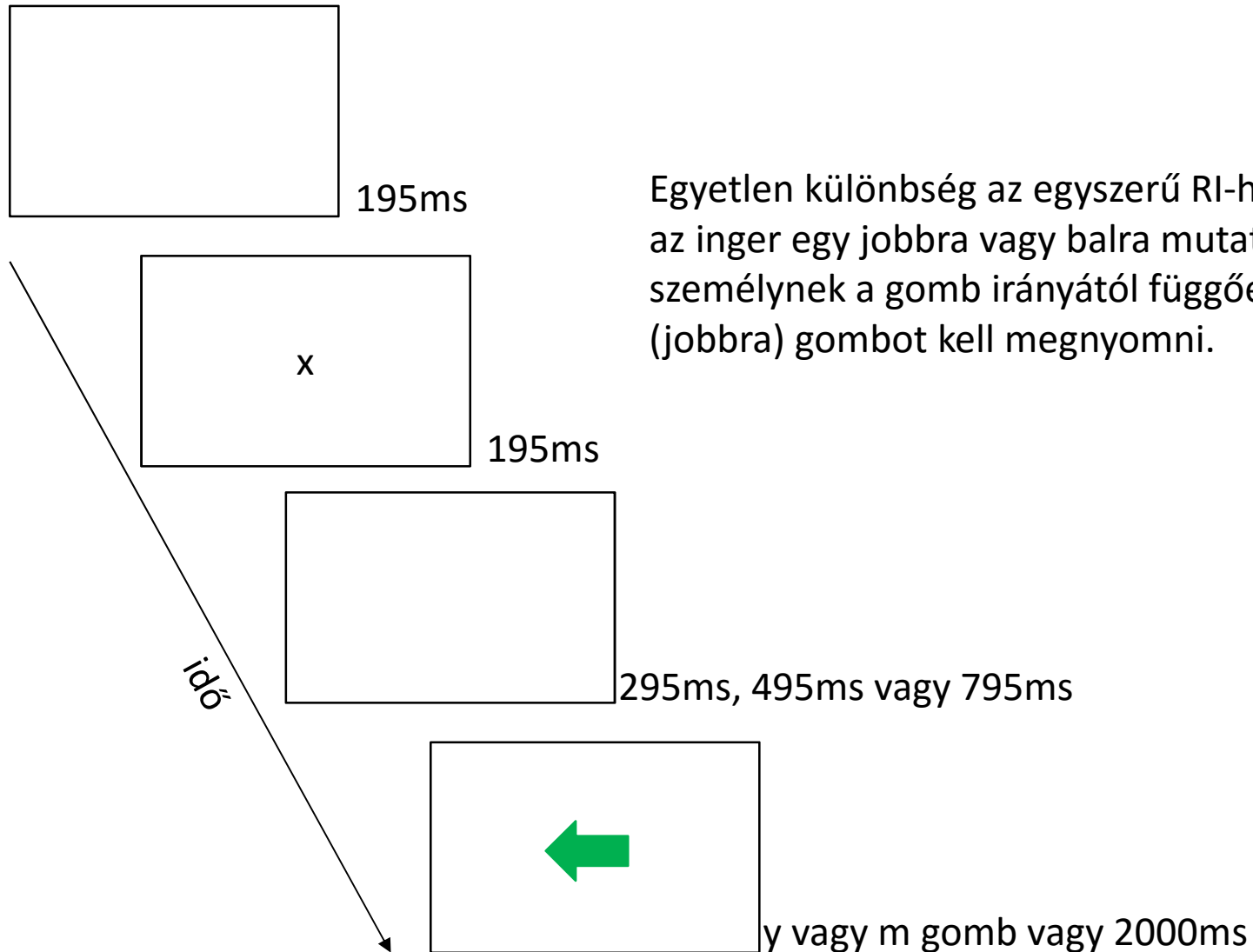
A búcsú elé nem mehet, mert akkor a összemosnánk a gyakorlást és a mérést. A loopba csak egyetlen blokkot lehet tenni, semmi mást. A blokkba végére nem mehet, mert akkor minden inger után lenne visszajelzés.



Helyette létrehozunk a kísérleti mérés körül egy újabb blokkot (nevezzük blokk_kísérlet-nek), ami a loop-ot és a visszajelzést fogja tartalmazni.

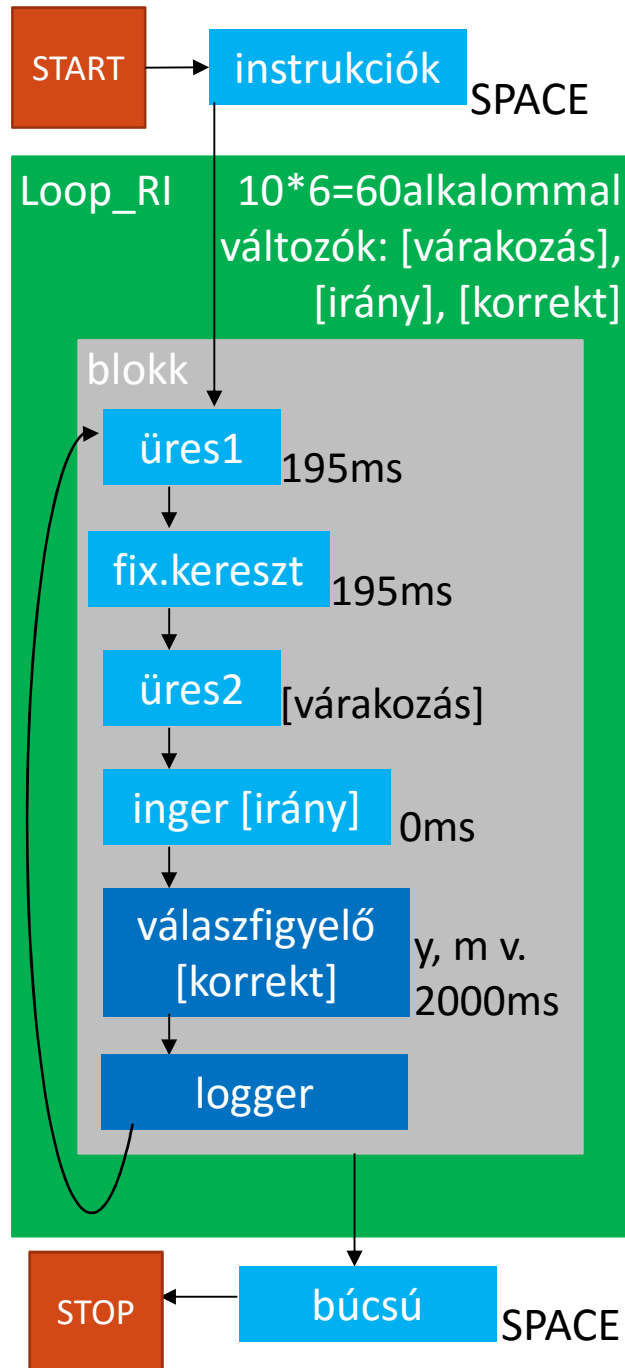


Választásos reakcióidő



Egyetlen különbség az egyszerű RI-hez képest, hogy most az inger egy jobbra vagy balra mutató nyíl, és a kísérleti személynek a gomb irányától függően az y (balra) vagy m (jobbra) gombot kell megnyomni.

Választásos RI

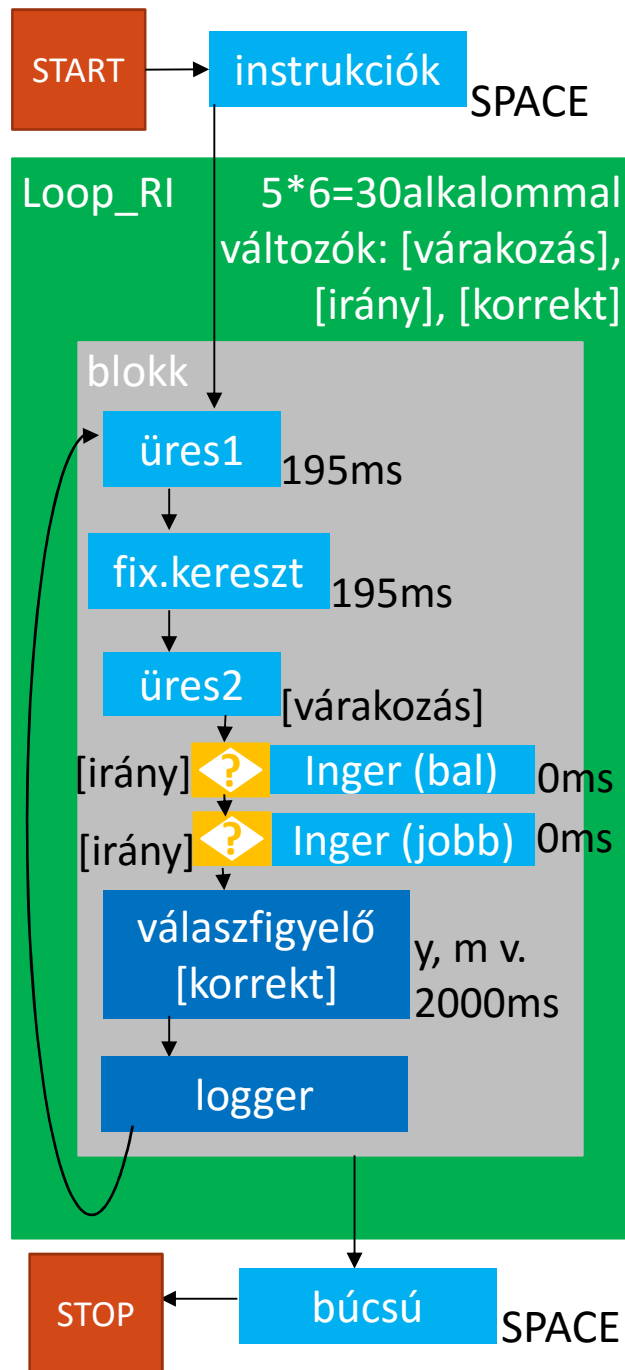


Loop_RI változó tábla

[irány]	[korrekt]	[várakozás]
bal	y	295
bal	y	495
bal	y	795
jobb	m	295
jobb	m	495
jobb	m	795

A változó táblánk most 3 változót tartalmaz. Az [irány] lehet jobb és bal, a [korrekt] lehet az iránynak megfelelően y és m, és van 3 késleltetési időnk is, Ez hat kombinációt jelent, hiszen mindkét irány minden késleltetéssel előfordulhat.

A [várakozás] változót az üres2 kapja meg, az irányt az inger, míg a [korrekt] választ a válaszfigyelő, így rögtön azt is ki tudja értékelni, helyes volt-e a ksz. válasza.



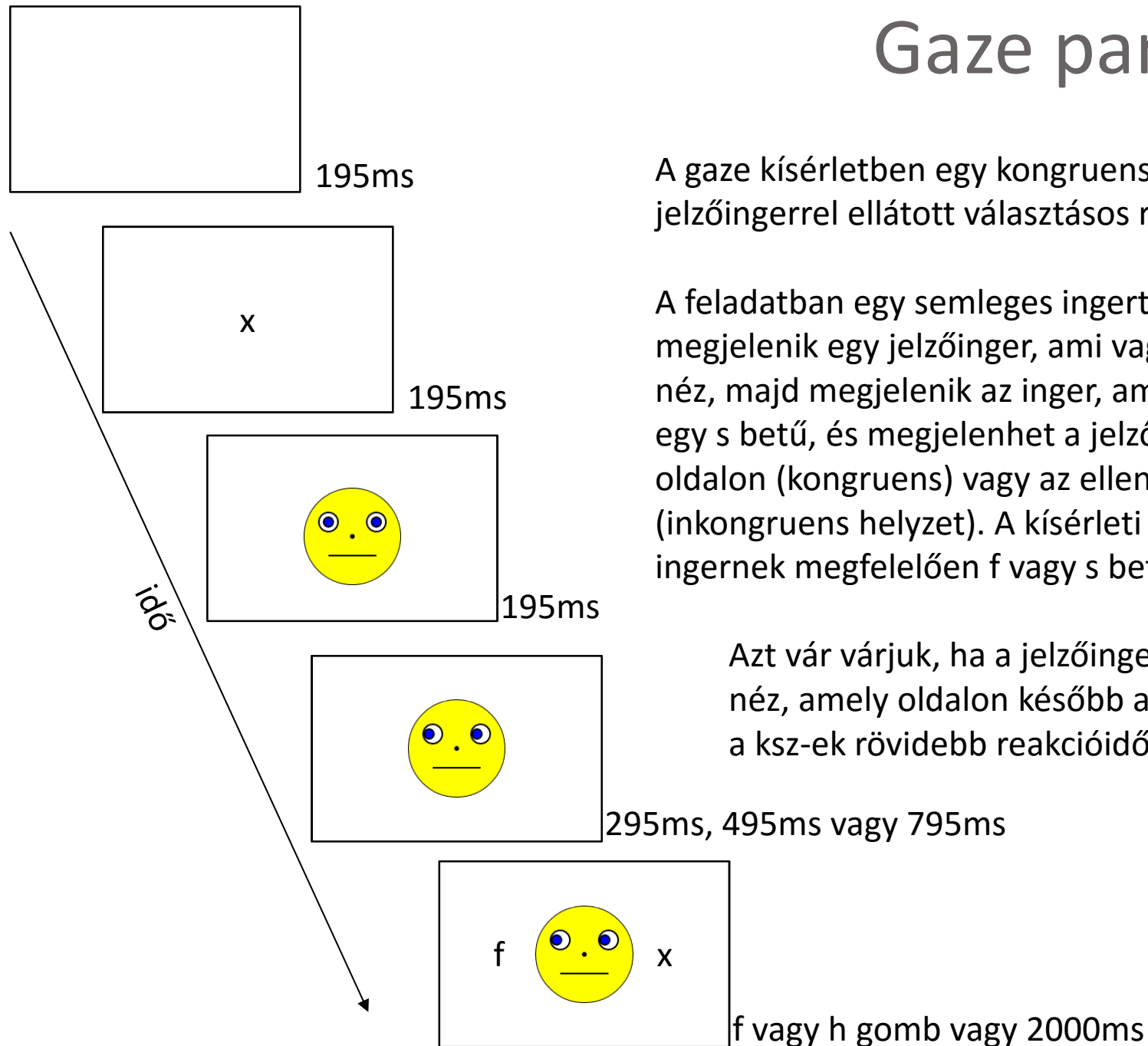
Választásos RI másik megoldás

Loop_RI változótábla		
[irány]	[korrekt]	[várakozás]
bal	y	295
bal	y	495
bal	y	795
jobb	m	295
jobb	m	495
jobb	m	795

A változótáblánk most 3 változót tartalmaz. Az [irány] lehet jobb és bal, a [korrekt] lehet az iránynak megfelelően y és m, és van 3 késleltetési időnk is, Ez hat kombinációt jelent, hiszen mindkét irány minden késleltetéssel előfordulhat.

A [várakozás] változót az üres2 kapja meg, az irányt az inger, míg a [korrekt] választ a válaszfigyelő, így rögtön azt is ki tudja értékelni, helyes volt-e a ksz. válasza.

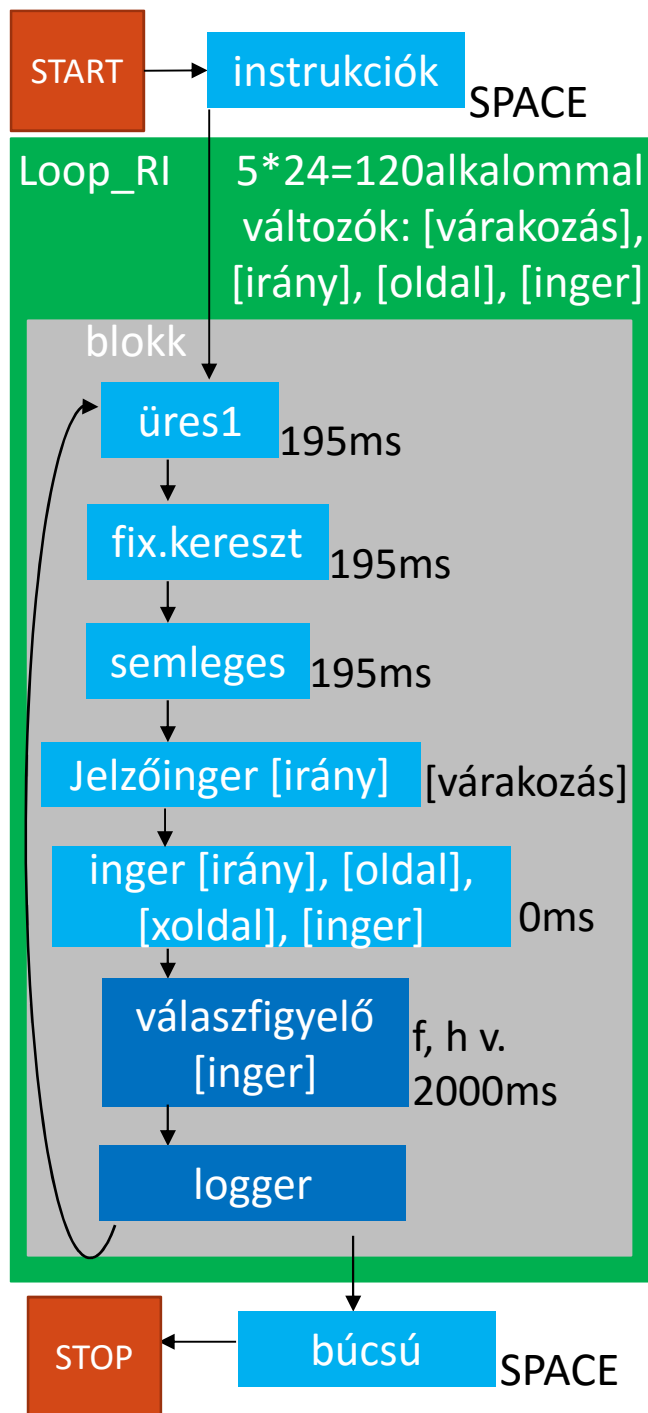
Gaze paradigm



A gaze kísérletben egy kongruens vagy inkongruens jelzőingerrel ellátott választásos reakcióidő feladat.

A feladatban egy semleges ingert követően megjelenik egy jelzőinger, ami vagy jobbra vagy balra néz, majd megjelenik az inger, ami lehet egy f vagy egy s betű, és megjelenhet a jelzőinger által jelzett oldalon (kongruens) vagy az ellentétes oldalon (inkongruens helyzet). A kísérleti személynek az ingernek megfelelően f vagy s betűt kell leütnie.

Azt vár várjuk, ha a jelzőinger abba az irányba néz, amely oldalon később az inger megjelenik, a ksz-ek rövidebb reakcióidővel fognak reagálni.



Loop_RI változótábla				
[irány]	[oldal]	[xoldal]	[inger]	[várakozás]
bal	-300	300	f	295
bal	-300	300	f	495
bal	-300	300	f	795
bal	-300	300	h	295
bal	-300	300	h	495
bal	-300	300	h	795
bal	300	-300	f	295
bal	300	-300	f	495
bal	300	-300	f	795
bal	300	-300	h	295
bal	300	-300	h	495
bal	300	-300	h	795
jobb	-300	300	f	295
jobb	-300	300	f	495
jobb	-300	300	f	795
jobb	-300	300	h	295
jobb	-300	300	h	495
jobb	-300	300	h	795
jobb	300	-300	f	295
jobb	300	-300	f	495
jobb	300	-300	f	795
jobb	300	-300	h	295
jobb	300	-300	h	495
jobb	300	-300	h	795

OpenSesame



Az Open-Sesame-ről

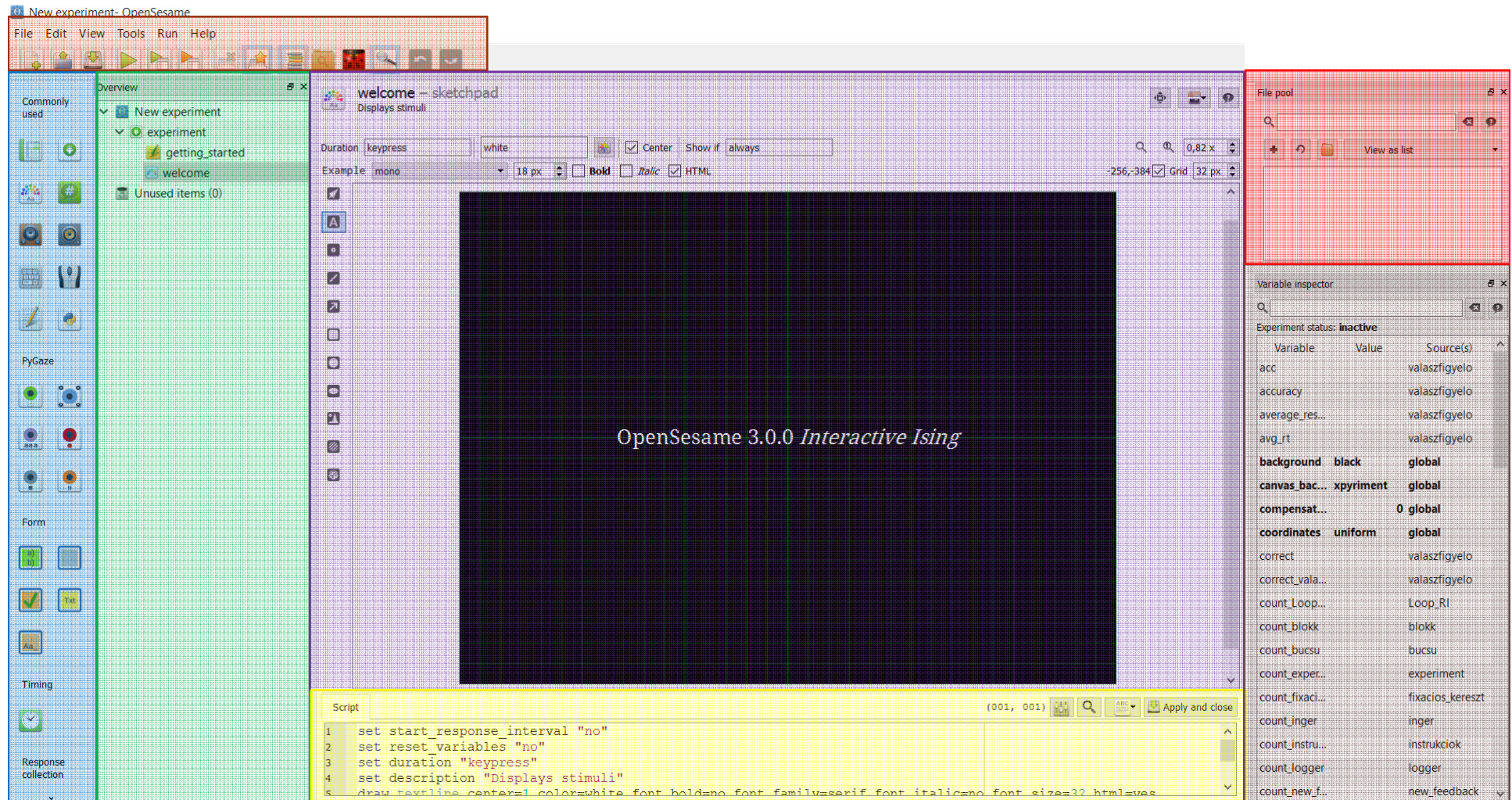


- Open-Sesame is a graphical, open-source experiment builder for the social sciences. It sports a modern and intuitive user interface that allows you to build complex experiments with a minimum of effort. With OpenSesame you can create a wide range of experiments. The plug-in framework and Python scripting allow you to incorporate external devices, such as eye trackers, response boxes, and parallel port devices, into your experiment.
OpenSesame is freely available under the General Public Licence.
- <http://osdoc.cogsci.nl/publications/>
- Ha használsz, őket írd be a publikációs jegyzékbe:
Mathôt, S., Schreij, D., & Theeuwes, J. (2012). OpenSesame: An open-source, graphical experiment builder for the social sciences. *Behavior Research Methods*, 44(2), 314-324. doi:10.3758/s13428-011-0168-7

Az Open-Sesame felülete

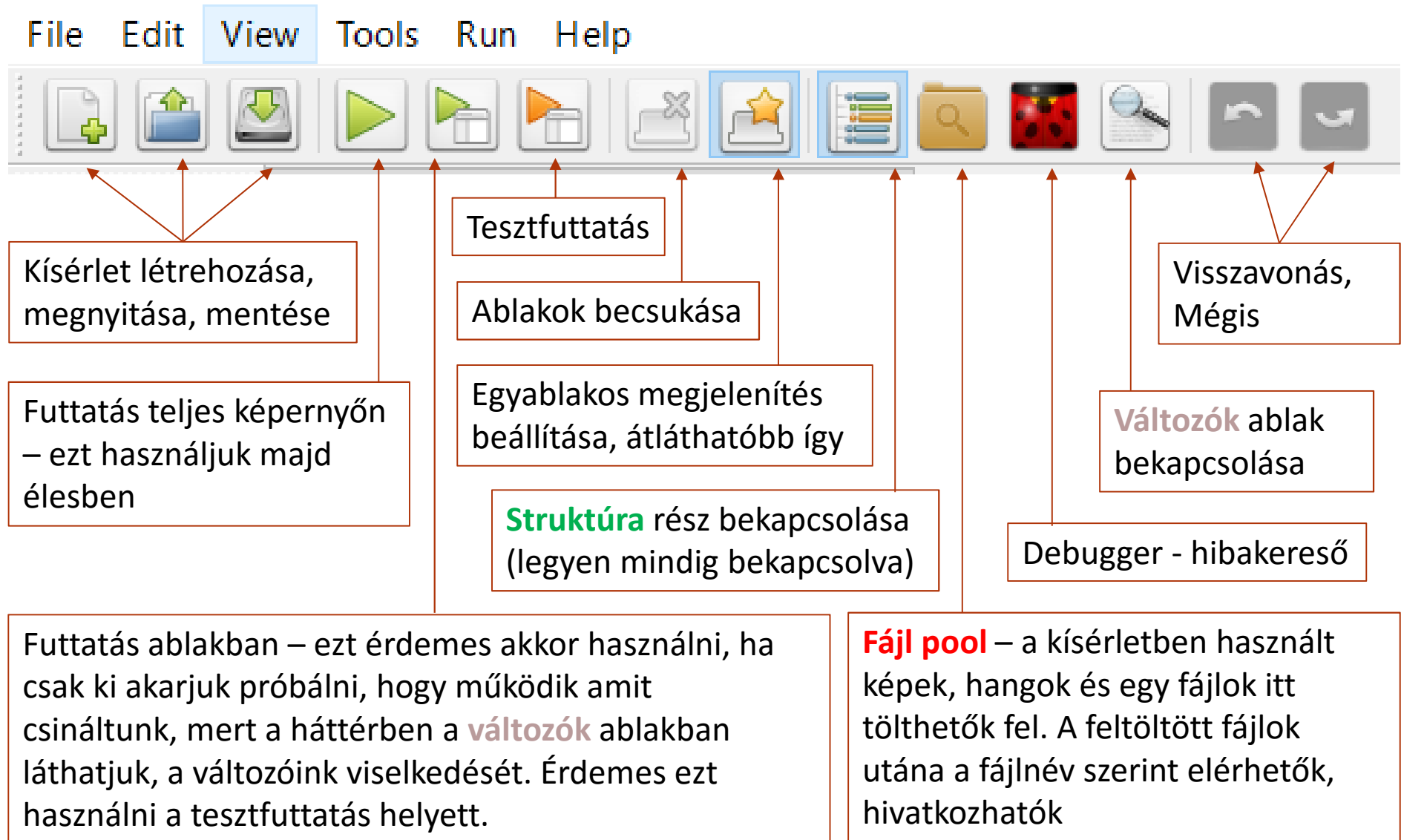
Az Open-Sesame szerkesztője az alábbi hét ablakból áll. Ezek közül megnyitáskor nincs mind bekapcsolva, de amikor odaérünk, megmutatom azt is, hol lehet előhívni őket.

Menüsáv, Eszközök, Struktúra, Szerkesztő, Szkript, Fájl pool, Változók

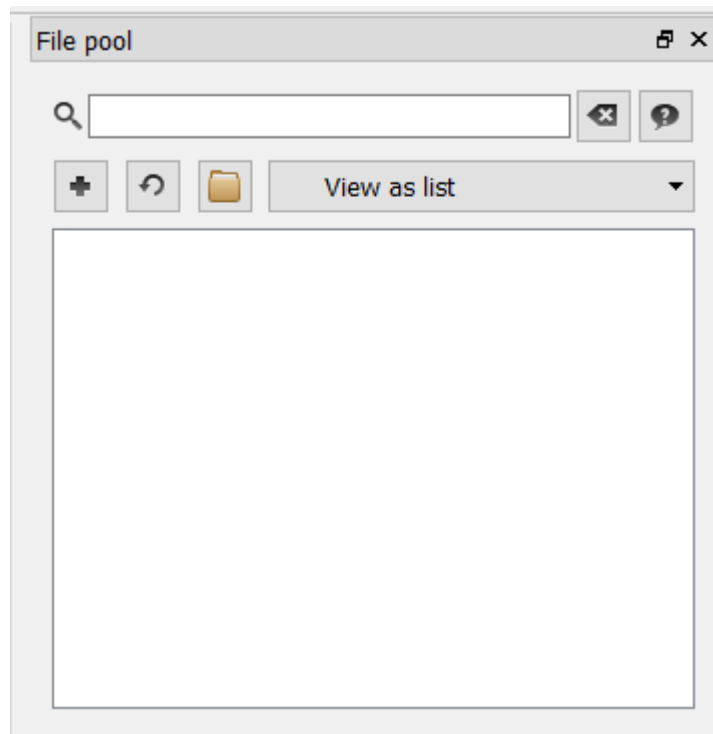


Menüsáv

Tartalmazza a megszokott menüket (fájl, szerkesztés, nézet...)



Fájl pool



- A kísérletben használt képek, hangok, videók és egyéb fájlok emelhetők be itt a plusz jelre kattintva.
- Érdemes megjegyezni, hogy a beolvasott fájlok részévé válnak az Open-Sesame fájlnak, így néhány nagyobb videó beemelésével jócskán megnőhet a programunk mérete.
- A nagy méretű fájlokat érdemes beolvasás helyett csak hozzálinkelni a kísérlethez. Ezt a sketchpadon lehet megtenni. Linkelés esetén figyelni kell rá, hogy amikor másik számítógépre feltesszük a programunkat, ne felejtsek otthon a linkelt állományokat sem.

Változók

Variable inspector

Experiment status: **inactive**

Variable	Value	Source(s)
background	black	global
canvas_bac...	xpyriment	global
compensat...	0	global
coordinates	uniform	global
count_exper...		experiment
count_getti...		getting_started
count_welc...		welcome
description	Default de...	global
foreground	white	global
height	768	global
response		welcome
response_ti...		welcome
response_ti...		welcome
response_w...		welcome
start	experiment	global
subject_nr	0	global
subject_pa...	even	global
time_experi...		experiment
time_gettin...		getting_started
time_welco...		welcome
title	New exper...	global
uniform_c...	yes	global
width	1024	global

- Szerkeszteni nem kell, de fontos átlátni, milyen változók generálódtak a kísérlet során, hiszen ezek lesznek **.csv**-ben is elmentve.
- Vastaggal vannak az **aktív változók** jelölve, melyeknek jelenleg értéke van, és hivatkozható, vékonygal az **inaktívak**, melyek a futtatás során csak máskor kapnak szerepet.
- A három oszlop a változók **nevét**, **aktuális értékét** és azt tartalmazza, a **struktúránk melyik eleméhez tartoznak**.
- Ha egy változó inaktív, akkor üres a hozzá tartozó érték mező, ha már aktív, de még nem kapott értéket, akkor az értéke **None**.
- Az egyszerű RI feladat tesztelése során figyeld meg, hogy a [várakozás] változónak hogyan változik az értéke! Keresd meg azt is, vajon melyik változó tárolja az ingerre adott válaszokat (SPACE gomb), és a válaszok reakcióidejét! Az Open-Sesame kiszámolja az átlagos reakcióidő adatot is, figyeld meg, hogyan változik az értéke a kísérlet alatt!

Eszközök

- Sokkal több eszköz/elem van, de ezeket használjuk legtöbbször.
Többiről itt olvashattok: <http://osdoc.cogsci.nl/2.9/usage/items/>

2. Loop

A loopban állítható be, hogy a benne lévő blokk hányszor és milyen változókkal fusson le. Egy loop csak szekvenciát (blokkot) tartalmazhat, és abból is csak egyet. A loopban állítjuk be a változókat is, tehát ide kerül a változótábla.

3. Sketchpad (vázlattömb)

Ezekbe valósítjuk meg, amit meg szeretnénk jeleníteni a képernyőn: instrukciók, ingerek, fixációs keresztek. Ide tudjuk a képeket, szövegeket betenni és szerkeszteni



1. Szekvencia (blokk)

A szekvencia elemek sorozatát tartalmazza sorrendben. Bármilyen elem csak szekvencia részeként tehető be. Például az RI kísérletben egy szekvencia a blokk, de egy szekvencia maga a kísérlet is.

4. Feedback (visszajelzés)

Ha végére értünk a kísérlet egy fázisának (pl. vége a gyakorlásnak), adhatunk visszajelzést a ksz-nek a teljesítményéről. Általában 2 értéket jelzünk vissza, az átlagos reakcióidőt és a helyes válaszok arányát.

Eszközök

- Sokkal több eszköz/elem van, de ezeket használjuk legtöbbször.
Többiről itt olvashattok: <http://osdoc.cogsci.nl/2.9/usage/items/>

7. Keyboard_response és mouse_response (válaszfigyelő billentyűzetről és egérről)
Egy inger kiadása után be kell gyűjtenünk a kísérleti személy választát. A ksz. választát attól a pillanattól figyeljük, hogy a válaszfigyelő betöltődött. A válaszfigyelőknek nincs konkrét megjelenési formája (nem változik meg a képernyőkép vagy a hang, ha átlépünk ebbe az elembe), ezért az inger megjelenését követően azonnal (0ms-ot követően) átléphetünk rájuk, az inger attól még a képernyőn fog maradni.

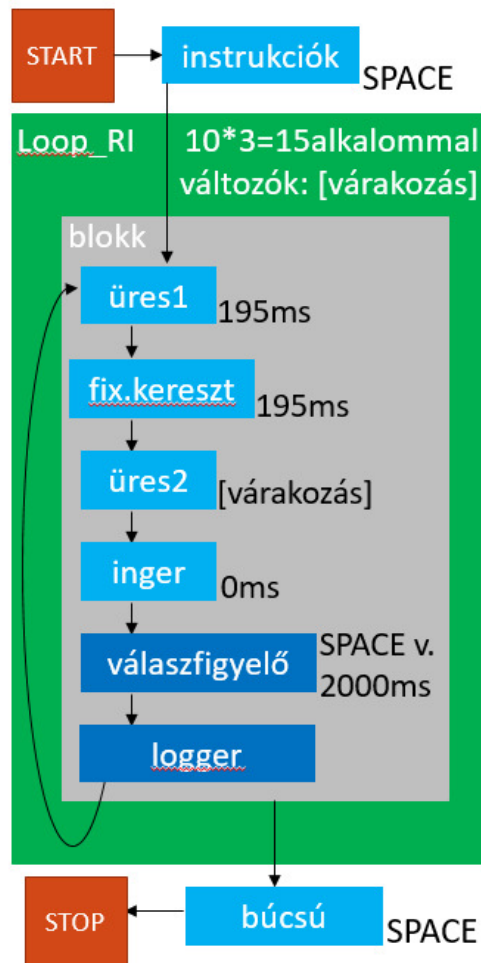
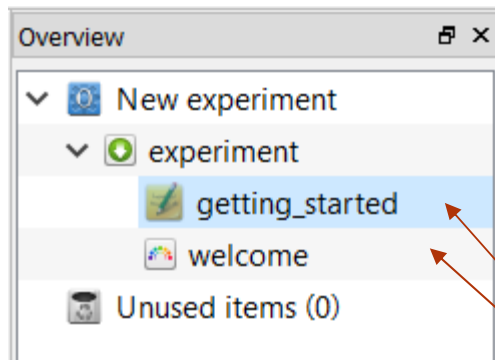


5-6. Sampler és synth (hanglejátszó és hangszintetizáló)
Hangjelzés adására két lehetőség is van, feltölthetünk egy előre elkészített hangot, és lejátszhatjuk azt, vagy generálhatunk hangjelzést a programon belül is.

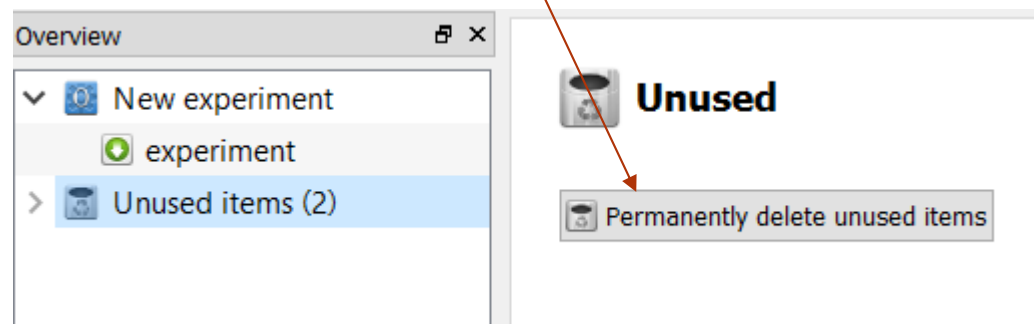
8. Logger
A választ el is kell menteni, ezért nagyon fontos, hogy minden válaszfigyelőt logger kövessen!

9. Inline script
Bonyolultabb kísérleteknél lehet, amit a GUIban nem lehet megoldani, hanem nekünk kell a kódot megírni. Itt lehet.

Struktúra



- A struktúrában fogjuk létrehozni a kísérlet vázát. A struktúra használatát egy későbbi példán részletesen bemutatom.
- Megnyitáskor a struktúra tartalmaz egy experiment blokkot, amiben két elem, egy getting_started comment elem és egy welcome sketchpad van.
- Ha már nincs szükségünk egy itemre, azt jobb klikk/törléssel törölhetjük.
- A törölt elemek a kukába (Unused items) kerülnek – ezek az elemek, ha már biztosan nincs rájuk szükségünk, akkor a kukába belelépve a végeleges törléssel eltávolíthatóak, vagy ha meggondoltuk, visszaállíthatóak.



Szerkesztő

- Szerkesztő felülete minden eszköznek máshogy néz ki.
- **1. LOOP**

Loop neve - itt is elnevezhetjük, de legtöbbször a struktúrában fogjuk (később)

Melyik elem van a loopban (a struktúrában fogjuk beállítani)

Ingerek bemutatási sorrendben vagy random sorrendben történjen

A kialakított loop (mérés) összegzése

The screenshot shows the 'Loop_RI - loop' configuration window. It includes fields for 'Item to run' (set to 'blokk'), 'Cycles' (set to 1), and 'Order' (set to 'random'). A status message indicates 'blokk will be called 1 x 1 = 1 times in random order'. At the bottom, there are buttons for 'Add variable', 'Rename variable', 'Remove variable', and 'Apply weights'. A pink box highlights the 'Add variable' button and the 'Apply weights' button, with a text box explaining the process of creating a variable table.

Itt kell kialakítani a változótáblát
Először add hozzá a változókat, add meg, hány sora van a változótáblának, egy sor hányszor forduljon elő a kísérletben, majd másold be a változótábla értékeit ide

Loop_RI változótábla

[várakozás]
295
495
795

Egy inger (sor) hányszor legyen bemutatva

A változótáblának hány sora van

Változó hozzáadása, átnevezése, törlése, súlyozása

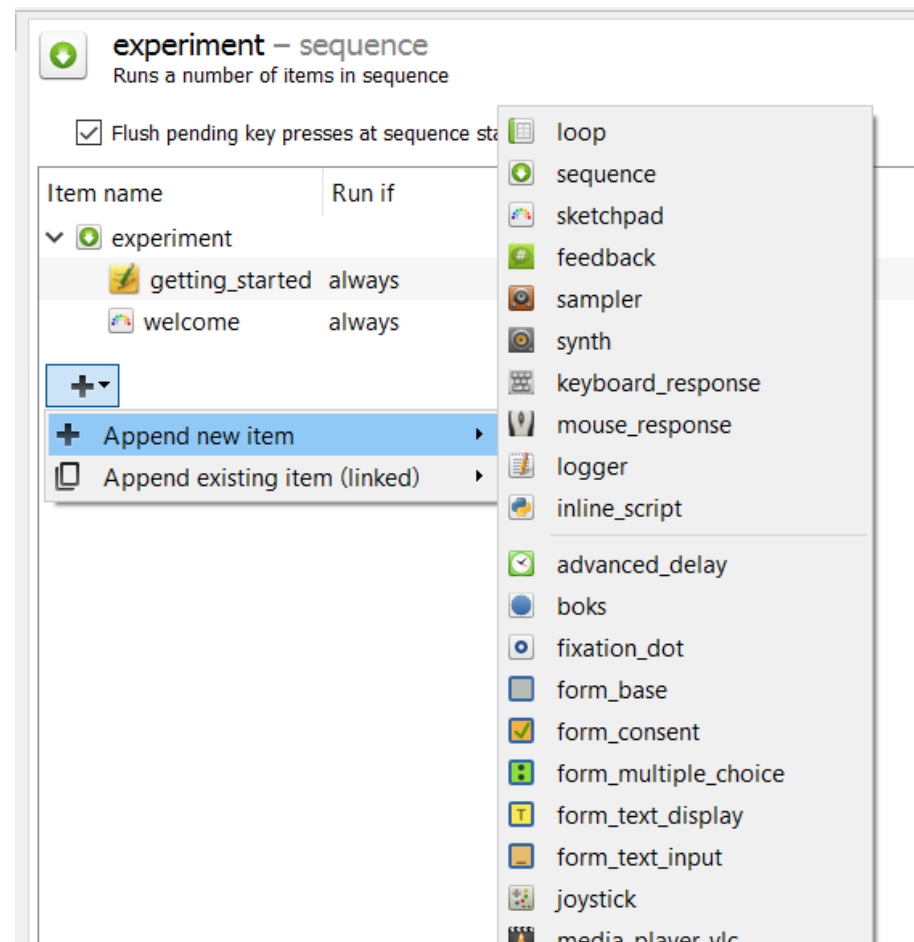
Szerkesztő

- **2. SEQUENCE**

- A sequenceben vannak felsorolva egy blokk elemei. Ha nem a struktúrába húztuk be az elemeket, akkor a pluszjelre kattintva itt is hozzáadható egy elem a blokkhoz. A ritkábban használt elemek is fel vannak sorolva.

- Itt változtatható meg az elemek sorrendje is húzással.

- Lehetőség van az ún. linkelt hozzáadásra is, ekkor egy már létező elemet adunk a szekvenciához (pl. egy második welcome sketchpadot). Innentől ezek össze lesznek kötve, és ha az egyik elemen változtatunk valamit, az a másikon is változni fog. Előnye, hogy csak egyszer kell megszerkeszteni a dolgokat, de ha nincs jól átgondolva a kísérlet, nagyon megnehezítheti az utólagos változtatásokat.



- **2. SEQUENCE**

- A Run if-ben beállíthatjuk azt is, hogy bizonyos elemek csak bizonyos feltételek teljesülése esetén jelenjenek meg.
- Például hangjelzést csak akkor akarunk küldeni a személynek, ha rossz választ adott. Ekkor fogjuk az Open-Sesame által generált változót, ami megnézi, hogy a válasz helyes volt-e, ez a [correct] változó, és a hangelem feltételeként beírjuk, hogy



[correct]==0

- Tehát az elem csak akkor fog lefutni, ha a correct változó értéke egyenlő nullával.

Szerkesztő

- 3. SKETCHPAD

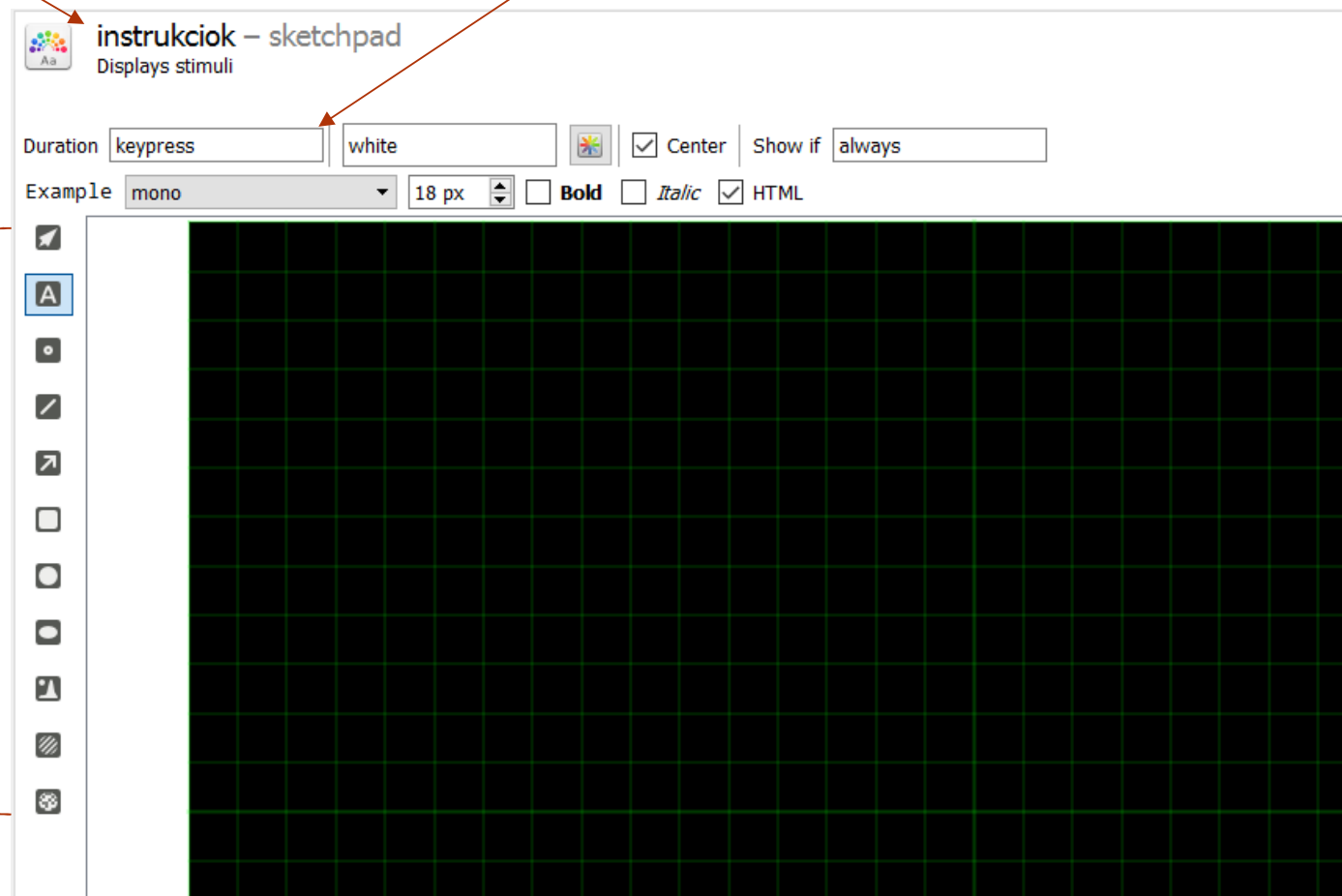
Duration: Meddig időzzünk az adott sketchpad-on

keypress: gombnyomásig

200: 200ms-ig

[várakozás]: a várakozás nevű változóban adott időhosszig

Sketchpad neve



Rajzeszközök
megj: ha előre
elkészített képet
akarunk beilleszteni,
azt először be kell
olvasni. A beolvasás
ablaka a menüsorból
nyitható meg

• 4. FEEDBACK

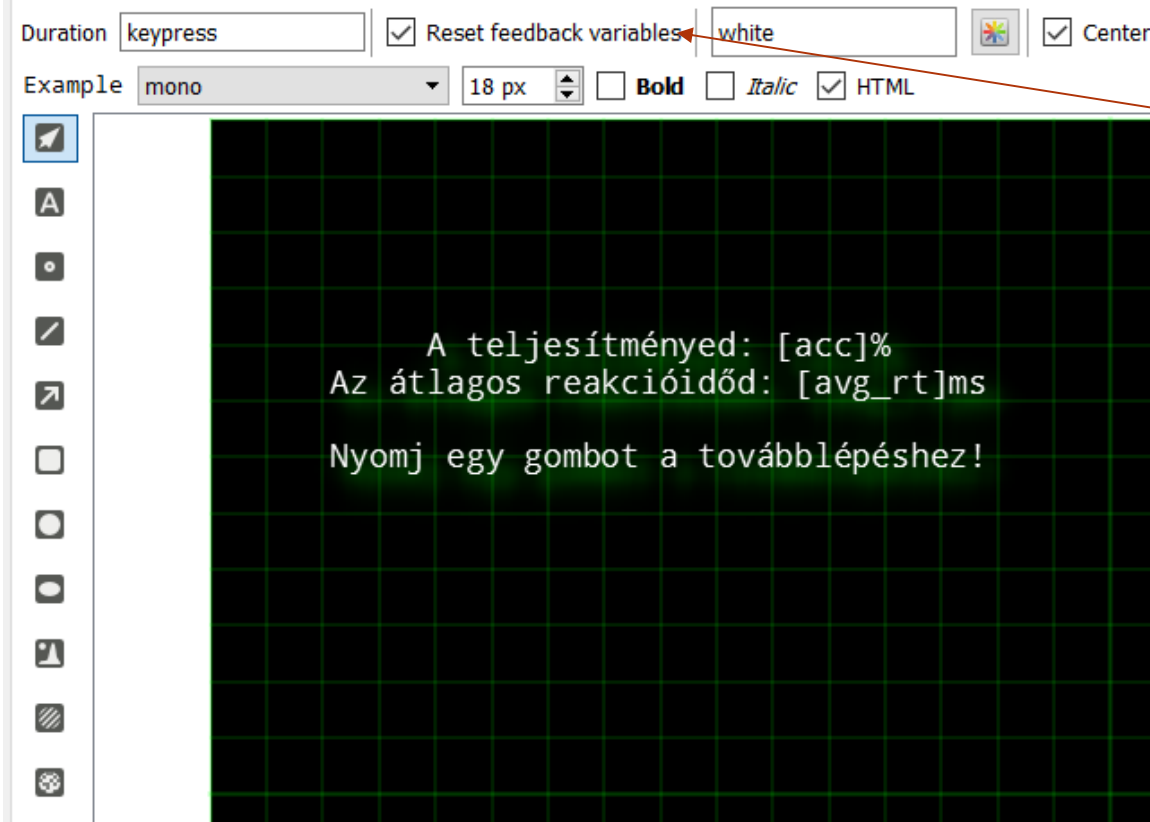
Az Open-Sesame automatikusan kiszámol néhány alap feedback változót:

acc – átlagos teljesítmény (helyes válaszok aránya)

avg_rt – átlagos reakcióidő

Ha visszajelzést akarunk adni, a feedbackben szöveges információként megadhatjuk ezeket.

A két változónevet szögletes zárójelbe kell tenni



Legtöbbször a visszajelzést követően resetelni szoktuk az acc és avg_rt értékét, így a következő visszajelzésbe már csak az ezt követő mérések fognak számítani

Minden más szempontból a feedback szerkesztőfelülete úgy néz ki, mintha sketchpad lenne.

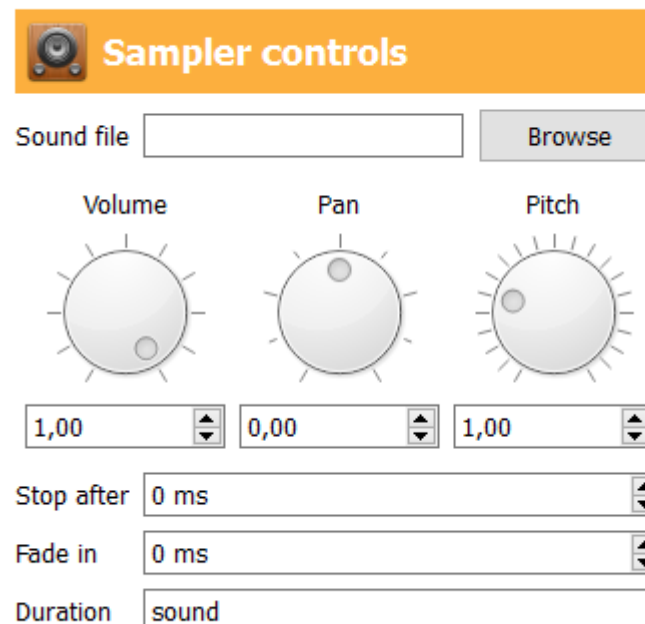
- **5. SAMPLER**

A sampler a poolba beolvasott hangfájlok lejátszója.

Ne felejtse el, hogy először be kell a poolba olvasni, majd a samplerben be kell a browse-zal hívni a fájlt.

Figyeljünk rá, hogy csak .wav és .ogg hangfájlok működnek!!

Beállíthatók a hangfájl tulajdonságai (hangerő, forrás oldala, hangmagasság), mennyit játszon le a hangfájlból, legyen-e beúsztatás, és meddig álljon a sampler elemen.



The image shows a software interface for a sampler. At the top is an orange header bar with a speaker icon and the text "Sampler controls". Below this, there is a "Sound file" label followed by a text input field and a "Browse" button. Underneath are three circular knobs labeled "Volume", "Pan", and "Pitch". Each knob has a corresponding numerical input field below it: "1,00" for Volume, "0,00" for Pan, and "1,00" for Pitch. At the bottom, there are three more input fields: "Stop after" with "0 ms", "Fade in" with "0 ms", and "Duration" with "sound".

- **6. SYNTH**

Nem csak kész hangfájlt használhatsz, hanem te is előállíthatod őket.

 **Synth controls**

Frequency
in Hertz (Hz) or by note, like 'A1'

Sine wave


Sawtooth wave


Square wave


White noise


Attack


Decay


Volume


Pan


Length

Duration

- 7.-8. Válaszfigyelők

Alapesetben billentyűzetről vagy egértől várunk választ.

A **reakcióidő** a response elem betöltődésétől a billentyű lenyomásáig vagy egérrel kattintásig eltelő idő, ezért az ingert megjelenítő sketchpad duration ideje mindig 0ms (azonnal továbblépünk a válaszfigyelésig)

A billentyű és egérválaszfigyelő nagyon hasonlóképp néz ki:

Correct response: mi lett volna a helyes válasz, ezt legtöbbször egy változóban tároljuk, és itt a változót szögletes zárójelben hivatkozunk meg (pl. [helyesvalasz])

Allowed responses: milyen választ fogadunk el: pl. ha a ksz-nek y vagy m gombot kell nyomni, akkor csak ezt a kettőt tekintjük válasznak, ha t betűt nyom, azt nem

Timeout: ha nem jön válasz, mennyi idő után lépünk tovább. Ekkor válaszként None-t mentünk el.

Flush: fontos, hogy bekattintva legyen! Amikor betöltődik a válaszfigyelő, a hozzá tartozó értékeket resetelni kell. Ha nem resetelünk, akkor a változók viszik tovább az előző mérésben vett értékeket, és ez csúnyán összekavarhatja az eredményeink.

 **new_keyboard_response** – keyboard response
Collects keyboard responses

Correct response

Allowed responses

Timeout

infinite

☒ Flush pending keypresses

 List available keys

 **new_mouse_response** – mouse response
Collects mouse responses

Correct response

Allowed responses

Timeout

infinite

☒ Visible mouse cursor

☒ Flush pending mouse clicks

• 9. Logger

A változókat minden a kísérlet szempontjából fontos változás után menteni kell (általában minden válasz után szokás)

A logger generál a csv fájlban egy új sort, és minden változó aktuális értékét elmenti.

Amíg tapasztalatlan vagy, érdemes minden változót elmenteni.

Ha tapasztaltabb vagy, a custom variable-lel kijelölhető, melyik változókat akarsz menteni (gondold át jól, nehogy a századik ember után jöjj rá, hogy valamit nem mentettél!)

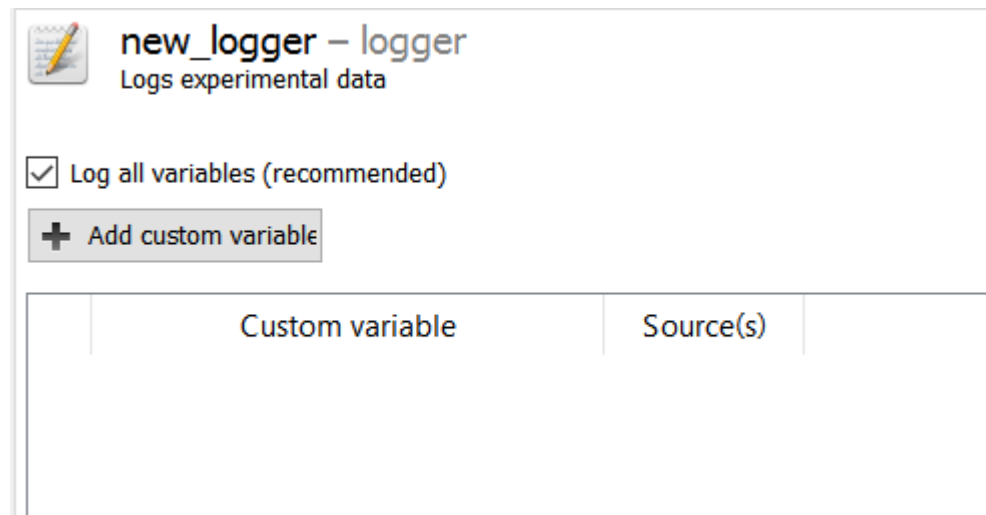
Logger által automatikusan generált hasznos változók:

response – előtte lévő elemre adott válasz

response_time – előtte lévő itemre adott válasz reakcióideje

acc – adott sorozatban az átlagos teljesítmény

avg_rt – adott sorozatban az átlagos reakcióidő



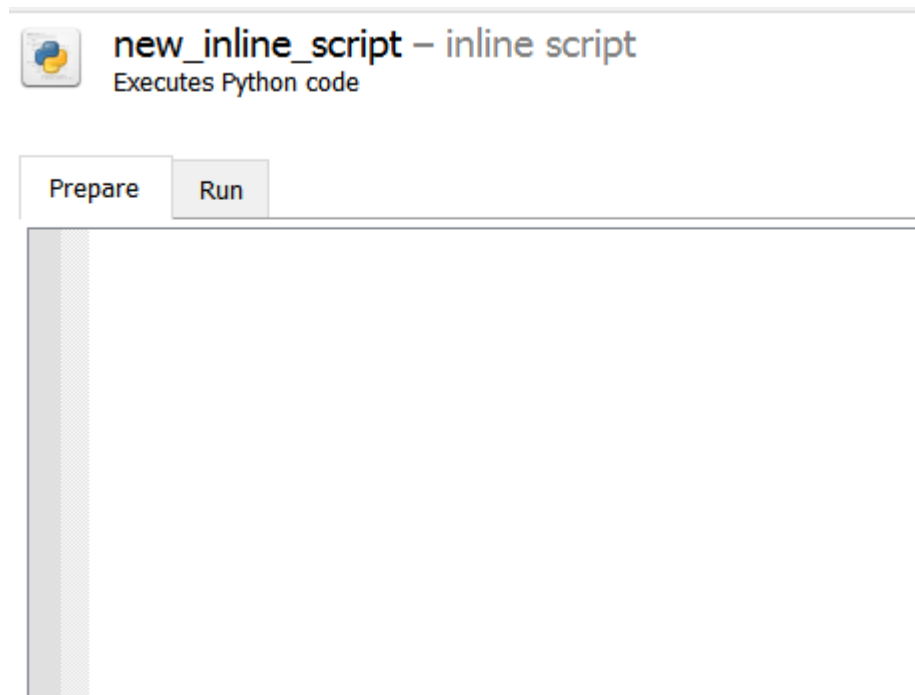
The screenshot shows a web interface for a logger. At the top, there is a title 'new_logger - logger' and a subtitle 'Logs experimental data'. Below this, there is a checkbox labeled 'Log all variables (recommended)' which is checked. Underneath the checkbox is a button labeled '+ Add custom variable'. At the bottom, there is a table with two columns: 'Custom variable' and 'Source(s)'. The table is currently empty.

Custom variable	Source(s)
-----------------	-----------

- **9. Inline script**

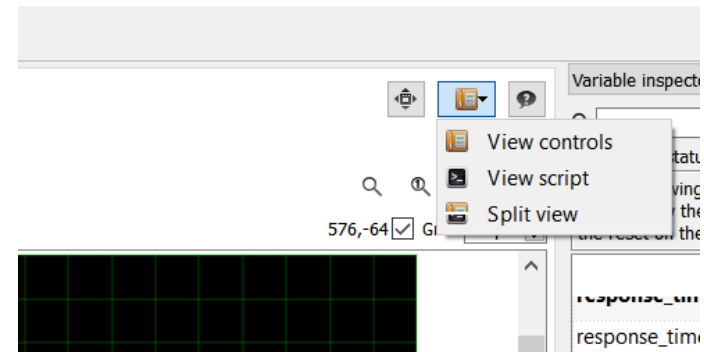
Ha valamit nem tudtál az Open-Sesame GUI-jában megvalósítani, Pythonban le is programozhatod.

Ilyet az órán nem fogunk csinálni.



Szkript

- Inline Python kódot nem fogunk írni, de néha szükségünk lehet néhány egyszerűbb szkriptre.
- Ezeket egy-egy elem szerkesztőjének jobb felső sarkában hozhatjuk elő. Nyissuk meg a welcome sketchpadot, és nyissuk meg a szkriptjét Split view-val!



```
Script (001, 001) [Icons] [Search] [ABC] [Apply and]

1 set start_response_interval "no"
2 set reset_variables "no"
3 set duration "keypress"
4 set description "Displays stimuli"
5 draw textline center=1 color=white font_bold=no font_family=serif font_italic=no font_size=32
  html=yes show_if=always text="OpenSesame 3.0.0 <i>Interactive Ising</i>" x=0 y=0 z_index=0
6
```

- A számunkra érdekes rész a draw textline két sora. Itt tehetjük a megjelenő szöveget, képet dinamikussá, egy változók értékétől függővé.
- Ilyenkor a változót szögletes zárójelbe kell tenni. Például van egy [inger] nevű változónk, ami a megjelenítendő szövegeket tartalmazza, akkor írhatjuk, hogy text=" [inger] "

Szkript

```
draw textline center=1 color=white font_bold=no font_family=serif font_italic=no font_size=32  
html=yes show_if=always text="OpenSesame 3.0.0 <i>Interactive Ising</i>" x=0 y=0 z_index=0
```

- show-if – feltételt adhatunk meg, milyen esetben jelenjen meg a szöveg
- text – megadhatjuk egy változóval, mi legyen a szöveg
- x és y – beállíthatjuk, a vízszintes és függőleges helyzetét (középről számítva)
- z-index – sorba rendezhetjük, hogy több inger esetén melyik legyen előrébb

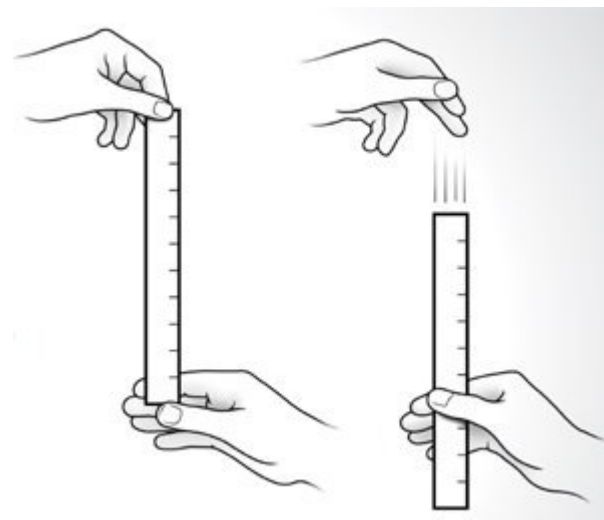
- Példa: tegyük fel, hogy van a következő változótáblánk, és szkriptünk:

```
draw textline center=1 color=white font_bold=no font_family=serif  
font_italic=no font_size=32 html=yes show_if=always text="[szoveg]"  
x="[vizszintes]" y="[fuggoleges]" z_index=0
```

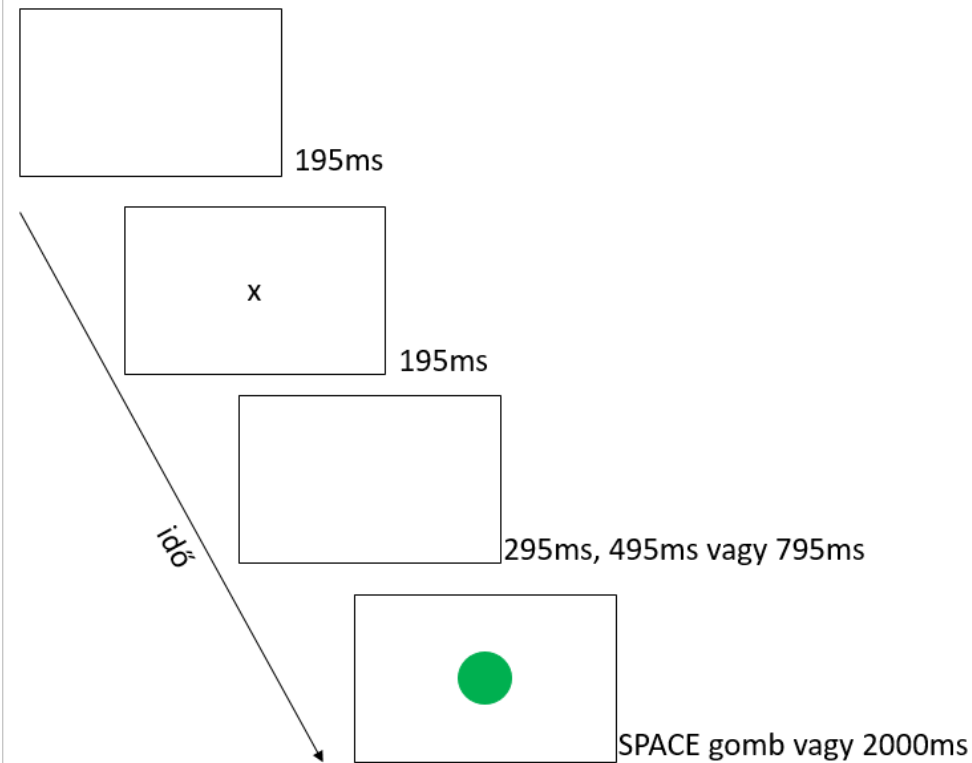
Loop_RI változótábla		
[szoveg]	[vizszintes]	[fuggoleges]
alma	300	-100
körte	300	-200
banán	300	-300
alma	-300	100
körte	-300	200
banán	-300	300

Ekkor az első inger az alma lesz, ami vízszintesen a középponttól +300 pixelre, függőlegesen -100 pixelre (tehát a középponttól jobbra, felfele) fog megjelenni

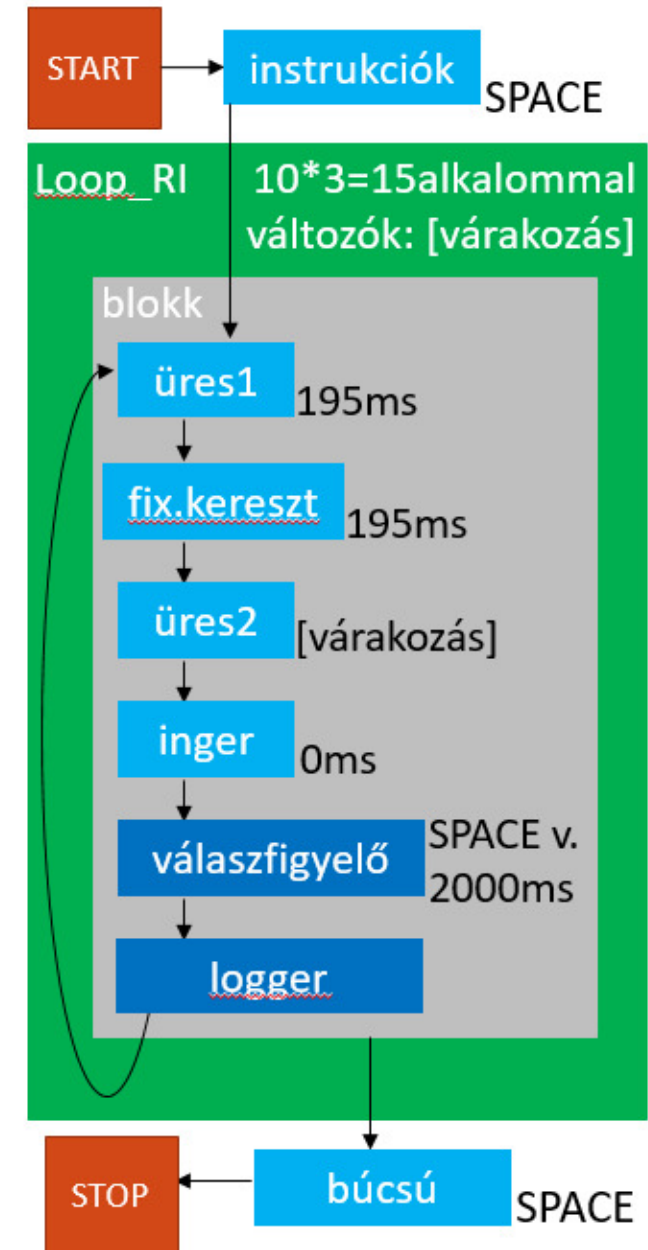
Példa: Egyszerű reakcióidő mérés



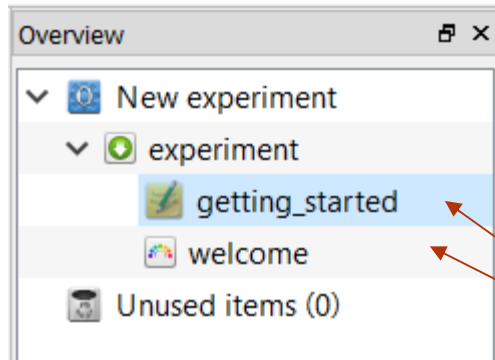
Emlékeztető



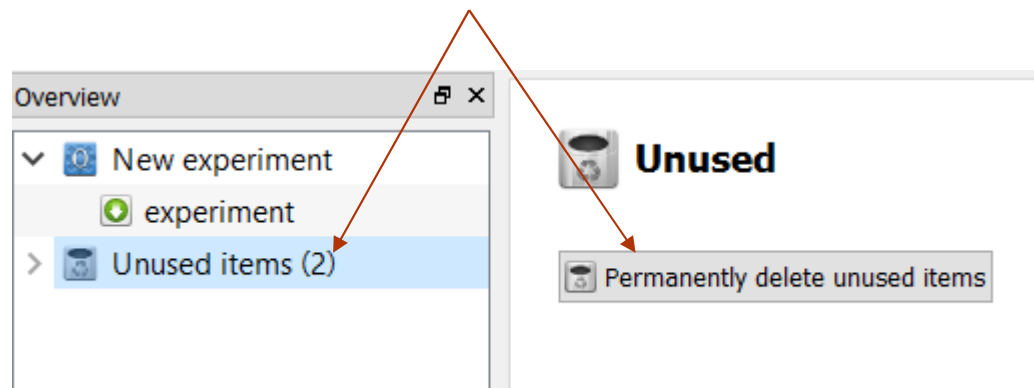
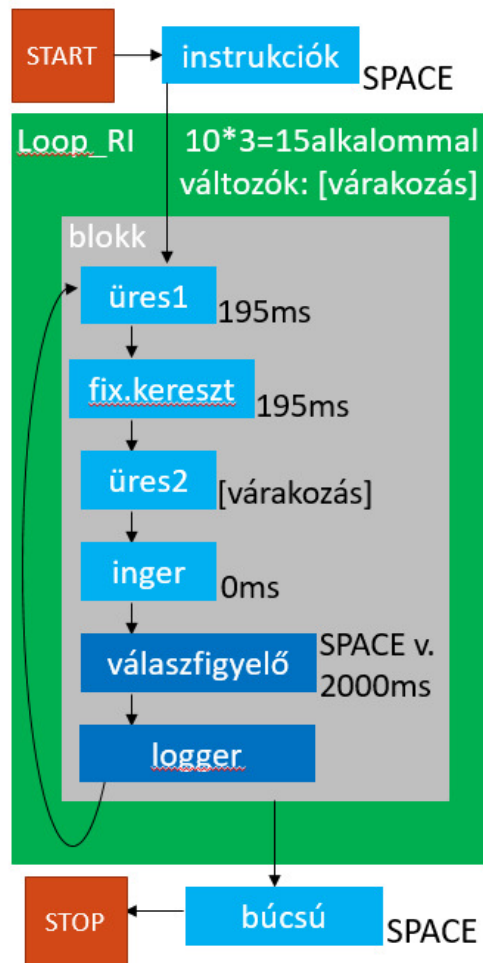
Loop_RI változótábla
[várakozás]
295
495
795

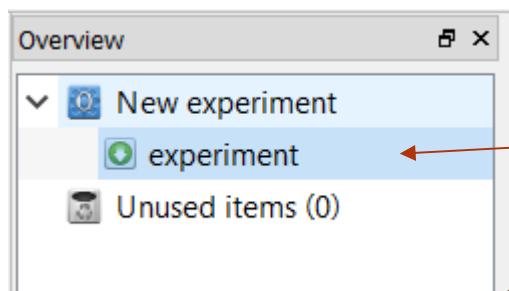


Alakítsuk ki a struktúrát!



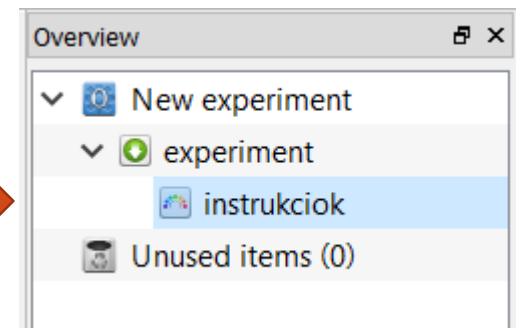
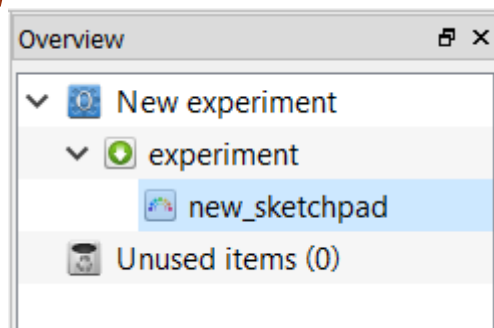
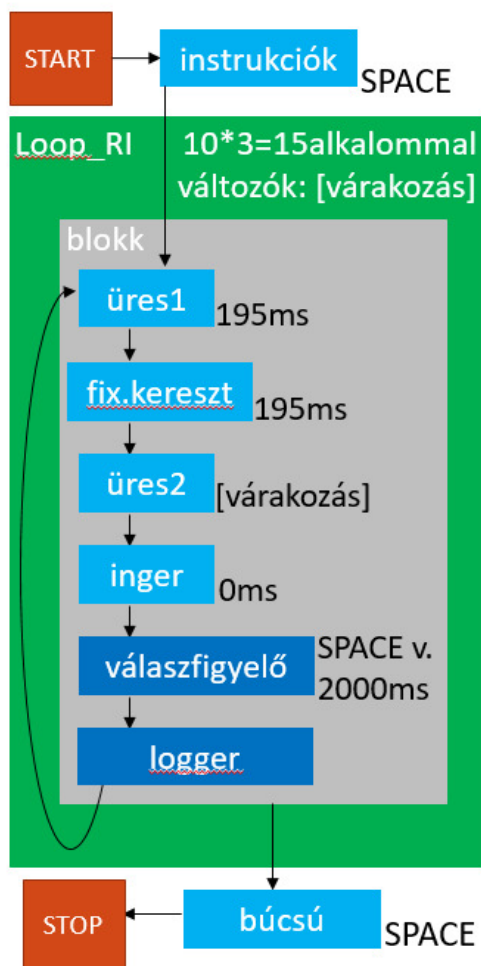
- Megnyításkor a struktúra tartalmaz egy experiment blokkot, amiben két elem, egy getting_started komment elem és egy welcome sketchpad van.
- A kommentre és az üdvözlő lapra nincs szükségünk, ezek törölhetők (jobb klikk, törlés)
- A törölt elemek a kukába (Unused items) kerülnek – ezek az elemek, ha már biztosan nincs rájuk szükségünk, akkor a kukába belelépve a végeleges törléssel eltávolíthatóak.



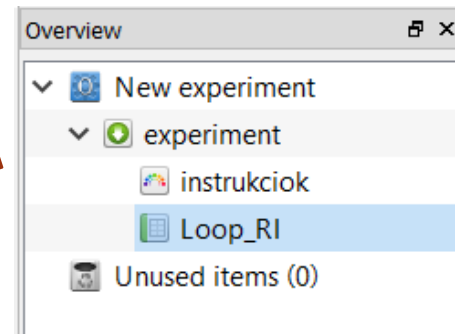
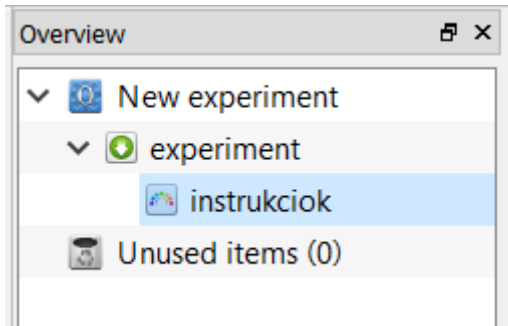


- Ha kitöröltük az üdvözlő szövegeket, akkor van egy üres kísérletünk. Kezdjük el feltölteni!

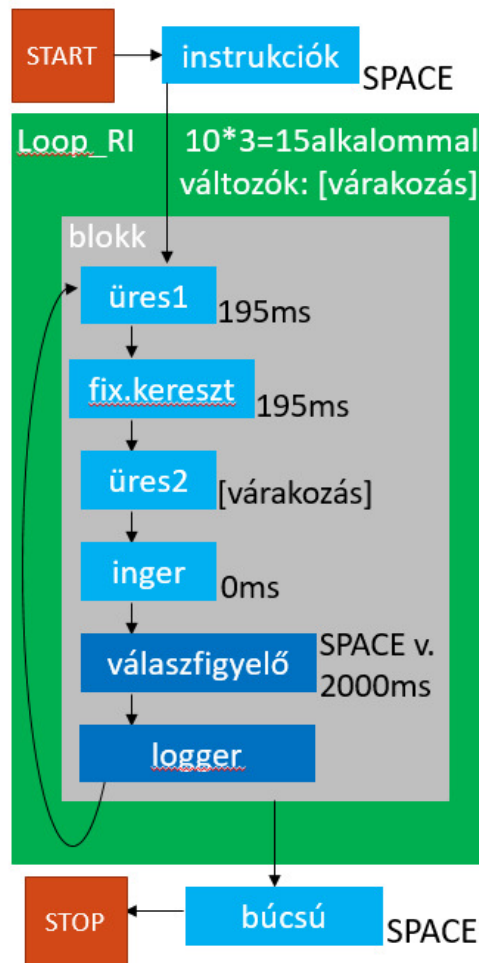
- Először kellenek instrukciók, ez szöveges infó, tehát sketchpadot használunk. Húzzunk bele az experiment szekvenciába egy sketchpadot az eszközök közül!



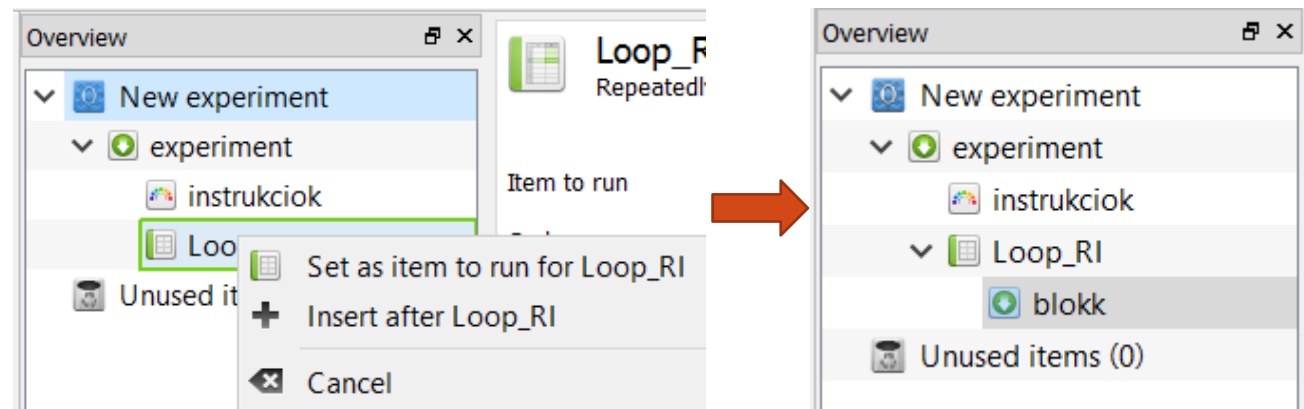
- Nevezzük el instrukcióknak! Figyeljünk rá, hogy ékezetes és speciális karaktereket valamint szóközöket nem lehet használni, így innentől minden ékezetek nélkül íródik majd (pl. instrukcio, ures), és a szóközöket alsóvonással helyettesítjük (pl. fixacios_kereszt)!

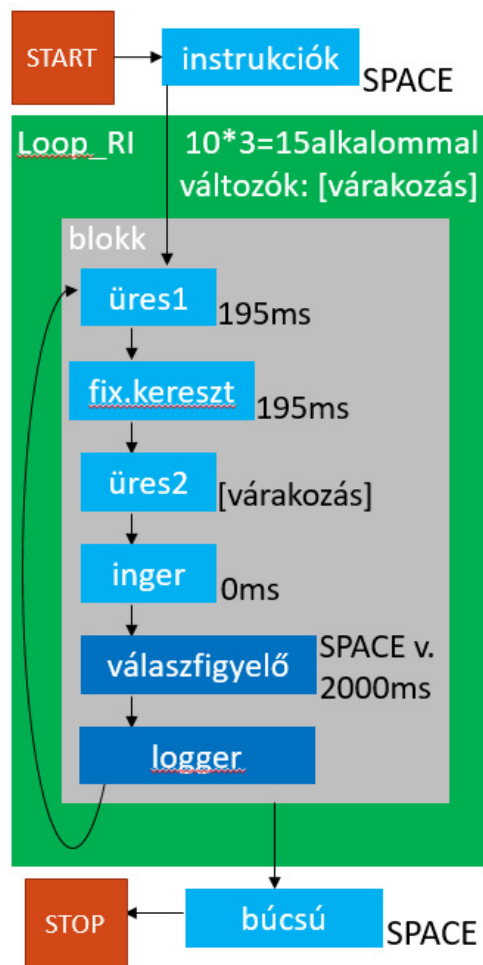
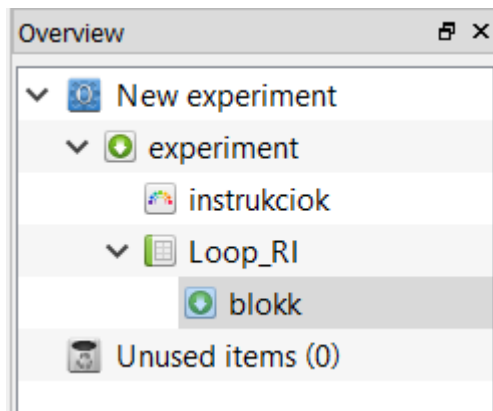


- Az instrukciók után jön a kísérleti loop, ehhez húzzunk egy loop elemet az instrukciók alá! Nevezzük el Loop_RI-nek.



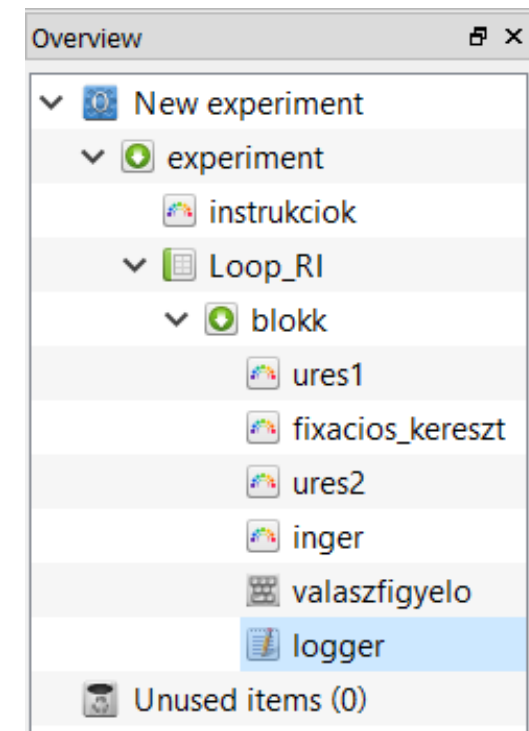
- A loop tartalmazza a reakcióidőmérés elemeinek blokkját, ami egy újabb szekvencia. Húzzunk be a loopba egy szekvenciát! A program megkérdezi, hogy a szekvenciát a loop részeként akarjuk használni, vagy a loop után szeretnénk tenni. Válasszuk azt, hogy a loop részeként! Nevezzük el ezt a belső szekvenciát blokknak!

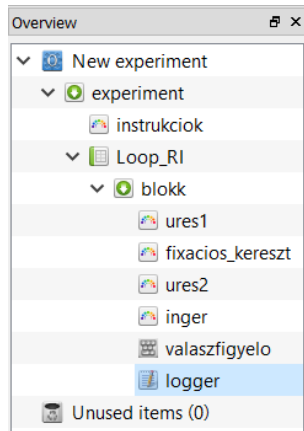




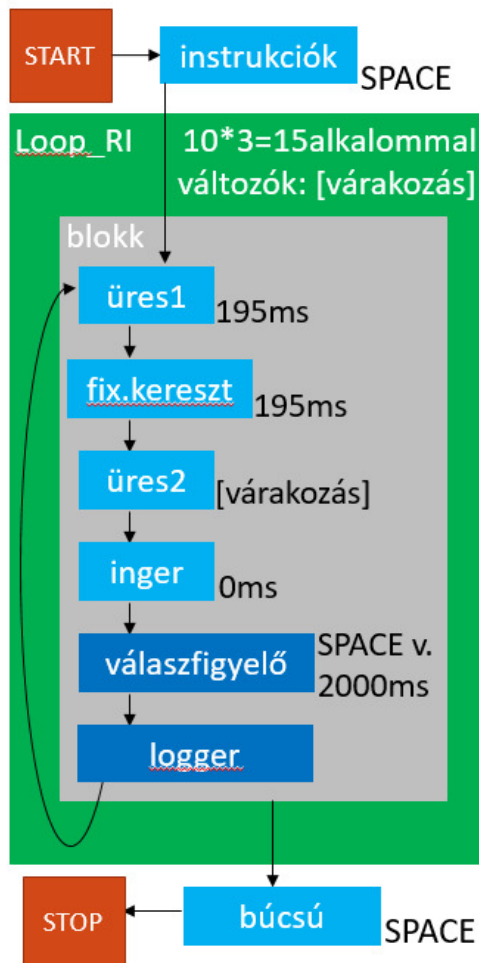
- A blokk tartalmaz egy üres képernyőt, egy fixációs keresztet, egy második üres képernyőt, egy ingert, egy válaszfigyelőt és egy loggert tartalmaz. Az üresek, a fix. kereszt és az inger egy-egy sketchpad lesz, ezeket adjuk hozzá a blokkhoz, és nevezzük el őket ures1, fixacios_kereszt, ures2 és inger néven!

- A válaszfigyelő egy keyboard_response elem lesz, míg a logger egy logger. Adjuk ezt is a blokkhoz, és nevezzük egy valaszfigyelo és logger néven!

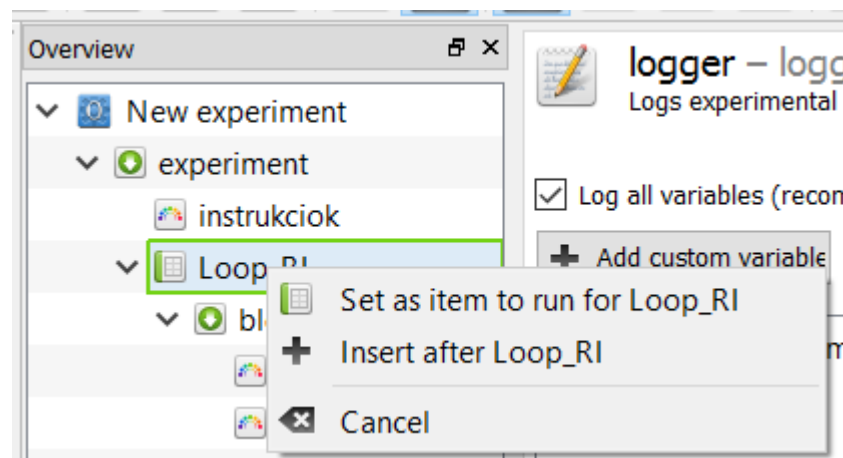




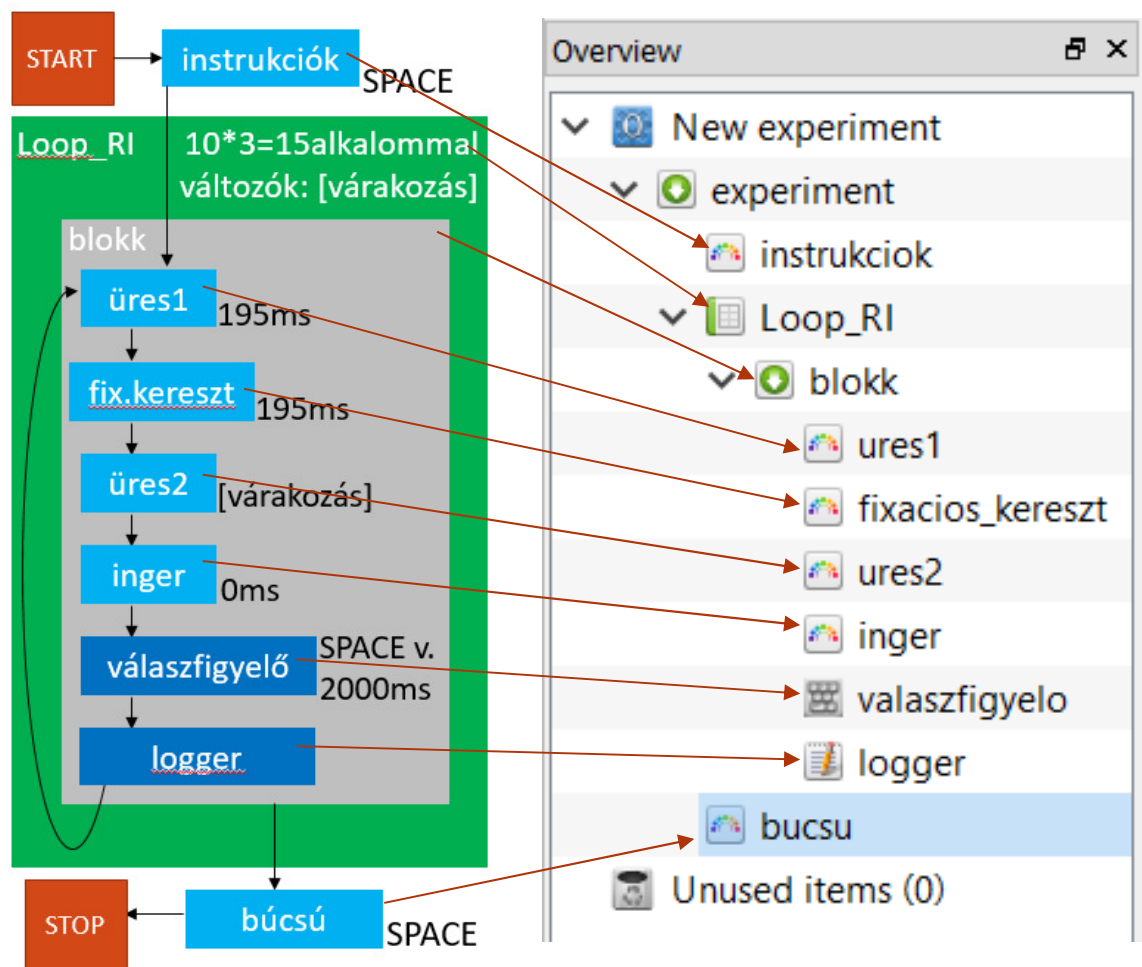
- Már csak egyetlen elem kell, egy búcsúzó sketchpad, aminek a loop UTÁN kell következni, de ha csak egyszerűen legalulra húzzuk, akkor a loop blokkjának végére kerülne be.



- Sok módszer van, hogyan lehet egy magasabb rendű sorozatot folytatni, később mutatok más módszert is, de most csináljuk úgy, hogy a sketchpadot húzzuk rá a loopra (mintha abba akarnánk elhelyezni), és a felugró ablakban válasszuk azt, hogy a loop után akarjuk elhelyezni, végül nevezzük el bucsunak!



- A struktúrával kész vagyunk, feleltessük meg egymásnak a két ábrát! Figyeljük meg, ahogy az Open-Sesame a behúzásokkal szemlélteti a hierarchikus struktúrát!



A felső szinten 3 elem van: az instrukciók, a loop és a búcsú.

A loop tartalmaz egy blokkot (mást nem is tud), és a blokk tartalmazza az üres1, fixkereszt, üres2, inger, válaszfigyelő és logger elemeket.

Állítsuk be az elemek tulajdonságait!

- 1. Instrukciók

instrukciók – sketchpad
Displays stimuli

Duration  ☒ Center Show if

Example 18 px ☐ Bold ☐ Italic ☒ HTML 384,410

Az instrukciók gombnyomásig maradnak fenn.

Kedves Kitöltő!

A következő tesztrel a reakcióidődet mérjük meg.
A képernyő közepén meg fog jelenni egy fixációs kereszt (fehér x),
mely jelzi, hogy készülj fel a feladatra. Erre NEM kell reagálnod.

Ezt követően meg fog jelenni egy zöld kör.
Amikor a zöld kör megjelenik, amilyen gyorsan csak tudod,
nyomd meg a SPACE (szóköz) gombot a billentyűzeten!

Ha felkészültél, a kísérlet indításához nyomd meg a SPACE billentyűt!

• 2. Loop_RI

- Az Add variable segítségével felvettél a várakozás változót.
- A Cycles-t 3-ra állítottam, mert a változótáblának 3 sora van, és a Repeat-et 10-re, mert minden várakozást 10-szer fogunk lemérni.
- Így összesen $3 \times 10 = 30$ mérés lesz a vizsgálatban.
- Végül beírtam a várakozás változó értékeit.

Loop_RI változótábla

[várakozás]
295
495
795

Loop_RI – loop
Repeatedly runs another item

Item to run: **blokk** Order: **random**

Cycles: **3** Repeat: **each cycle 10,00 time(s)**

blokk will be called $3 \times 10 = 30$ times in **random** order

☐ Show advanced options

+ Add variable **Rename variable** **Remove variable** **+ Apply weights** **Variable wizard**

varakozas	
1	295
2	495
3	795

- 3. blokk
- Itt nincs mit beállítanunk 😊

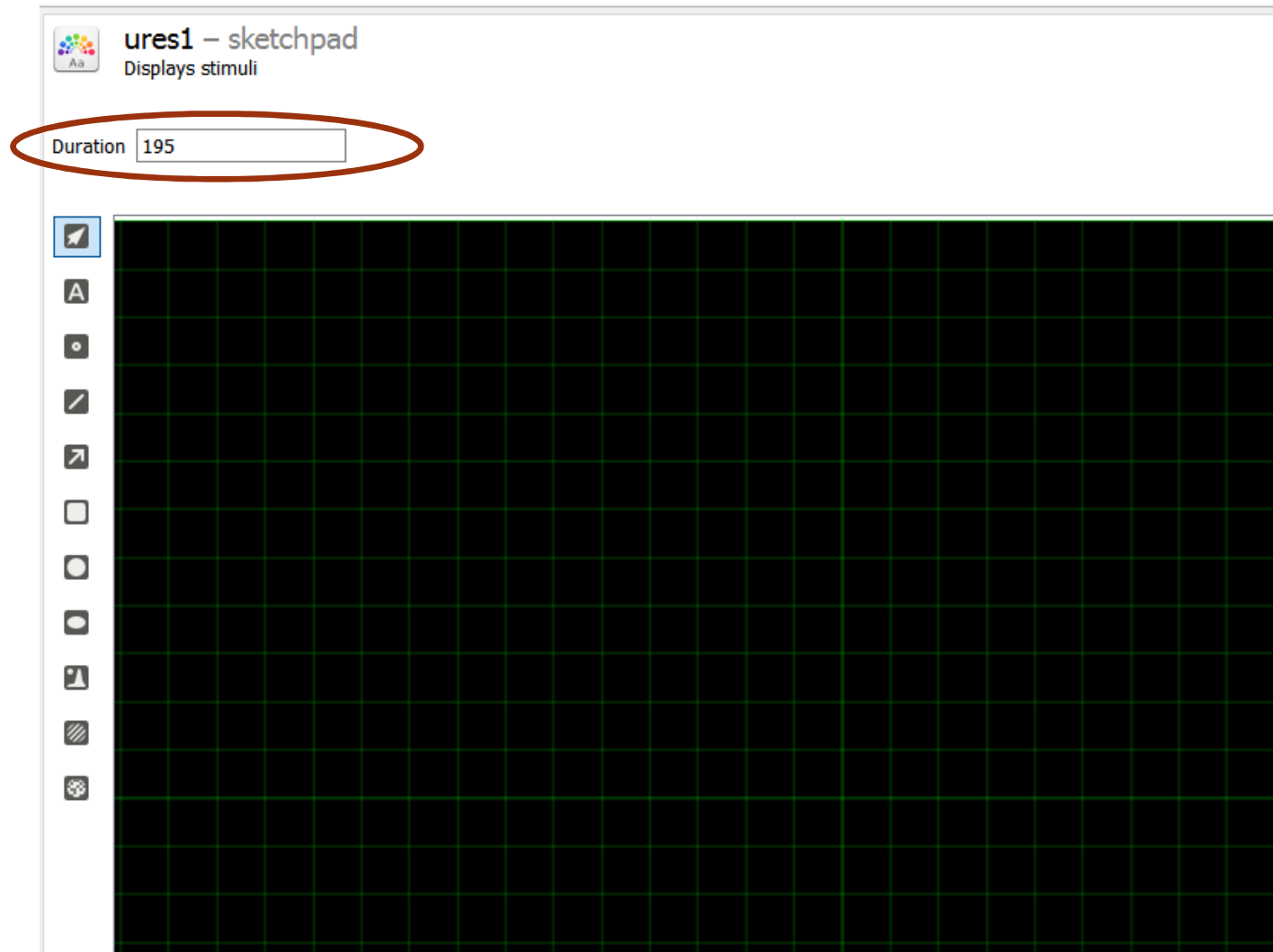
 **blokk – sequence**
Runs a number of items in sequence

☒ Flush pending key presses at sequence start

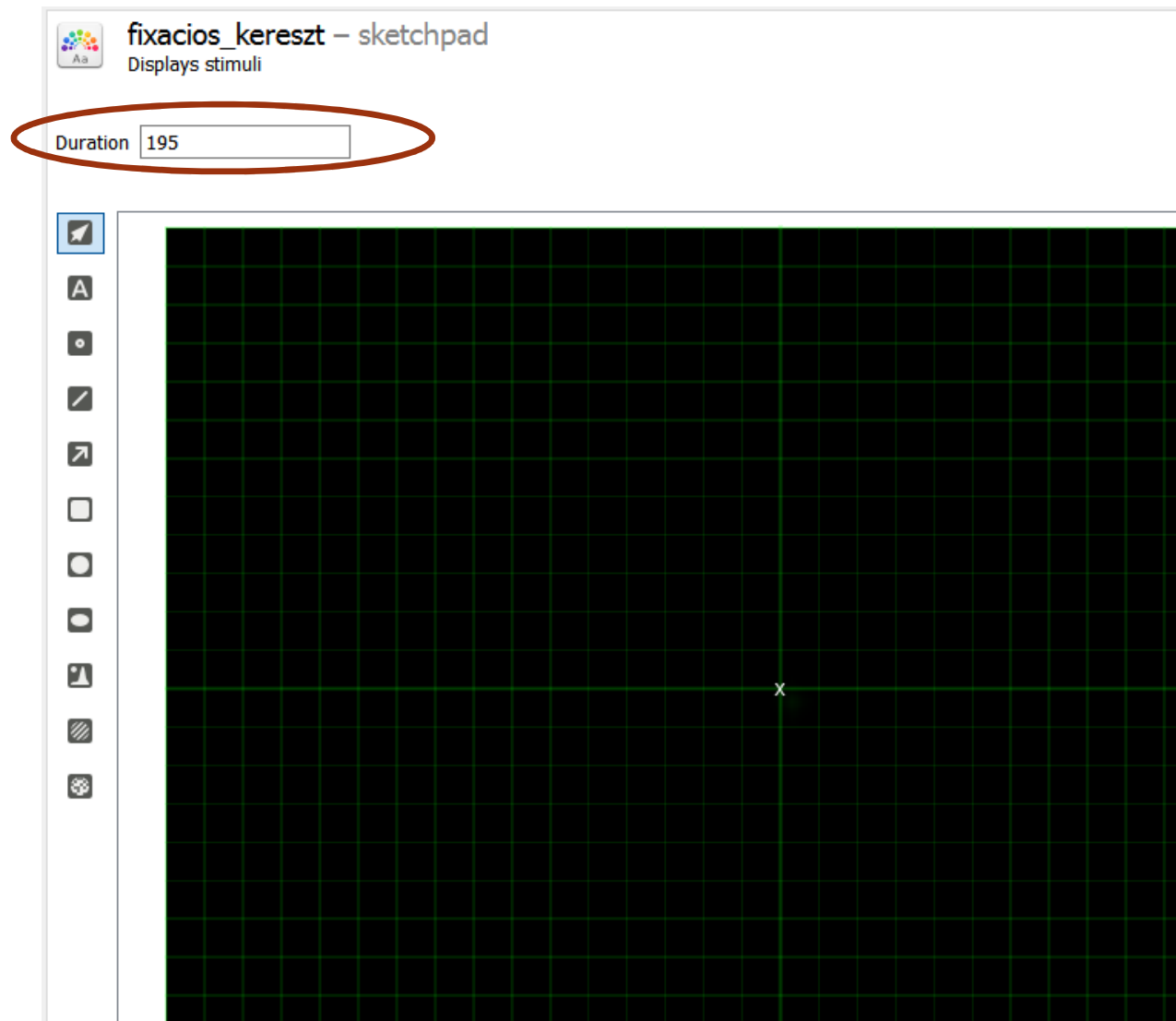
Item name	Run if
▼  blokk	
 ures1	always
 fixacios_kereszt	always
 ures2	always
 inger	always
 valaszfigyelo	always
 logger	always



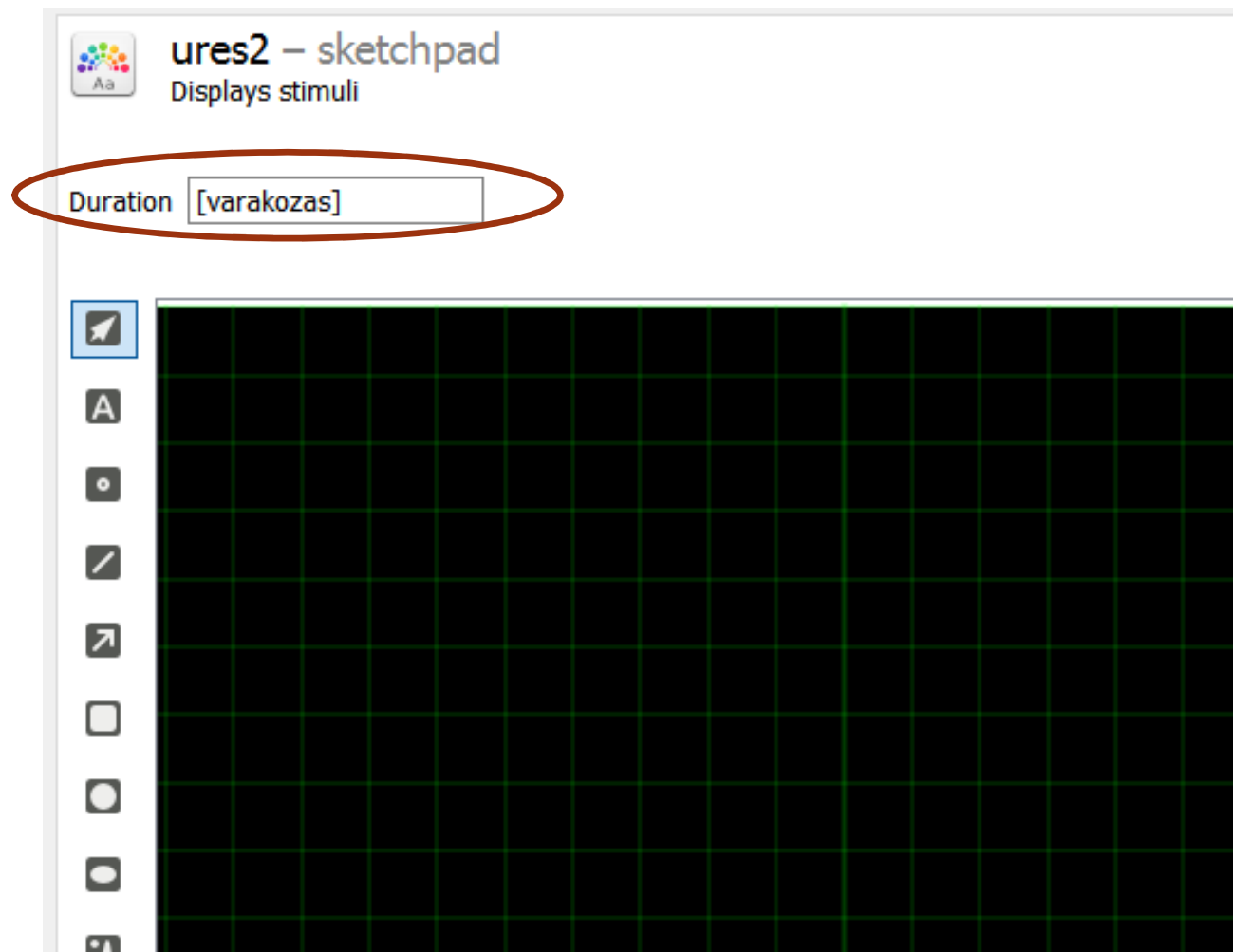
- 4. üres1
- Ez egy üres képernyő.
- Egyetlen dolgot kell beállítanunk, hogy meddig legyen látható: 195ms-ig.



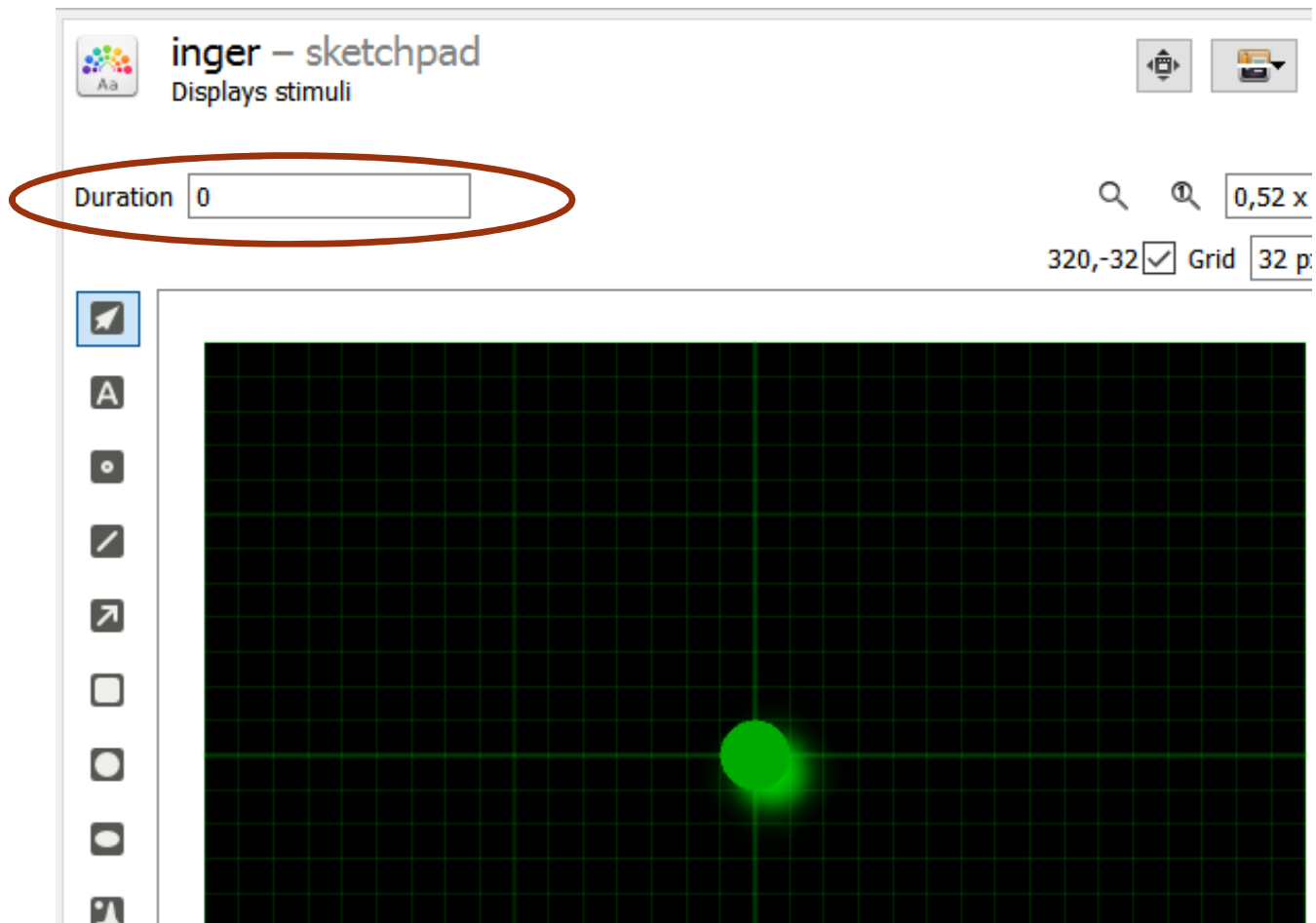
- 5. fixációs kereszt
- A képernyő közepére tegyünk egy x-et! Állítsuk be, hogy 195ms-ig legyen látható!



- 6. üres2
- A második üres képernyő a kísérlet alatt eltérő hosszúságig lesz látható (295ms, 495ms és 795ms). A késleltetés hosszát a [varakozas] változóban tároltuk. Most adjuk meg, hogy az üres2 hossza [varakozas] legyen!



- 7. inger
- Az inger egy zöld kör lesz, amit a képernyő közepére teszünk (ne ijedj meg, az árnyékolás a kísérletben nem lesz ott)
- Az inger hossza 0ms lesz, hiszen a megjelenését követően azonnal továbblépünk a válasz figyelésére.



- 8. válaszfigyelő

- Most nincs értelme a helyes válasz mezőnek, mert nincs döntési helyzet a feladatban, a ksz-nek mindig a SPACE billentyűt kell megnyomnia.
- Az elfogadható válasznak írjuk be, hogy SPACE-t várunk
- Ha 2000ms alatt (2másodperc) nem válaszol a ksz, akkor lépünk tovább a következő mérésre. Ekkor a válaszaként None lesz elmentve.
- Flush be van kapcsolva: a válaszfigyelő betöltésekor az előző mérés értékeit töröljük

 **valaszfigyelo – keyboard response**
Collects keyboard responses

Correct response

Allowed responses

SPACE

Timeout

2000

☒ Flush pending keypresses

 List available keys

- 9. logger
- Itt nincs mit beállítanunk 😊

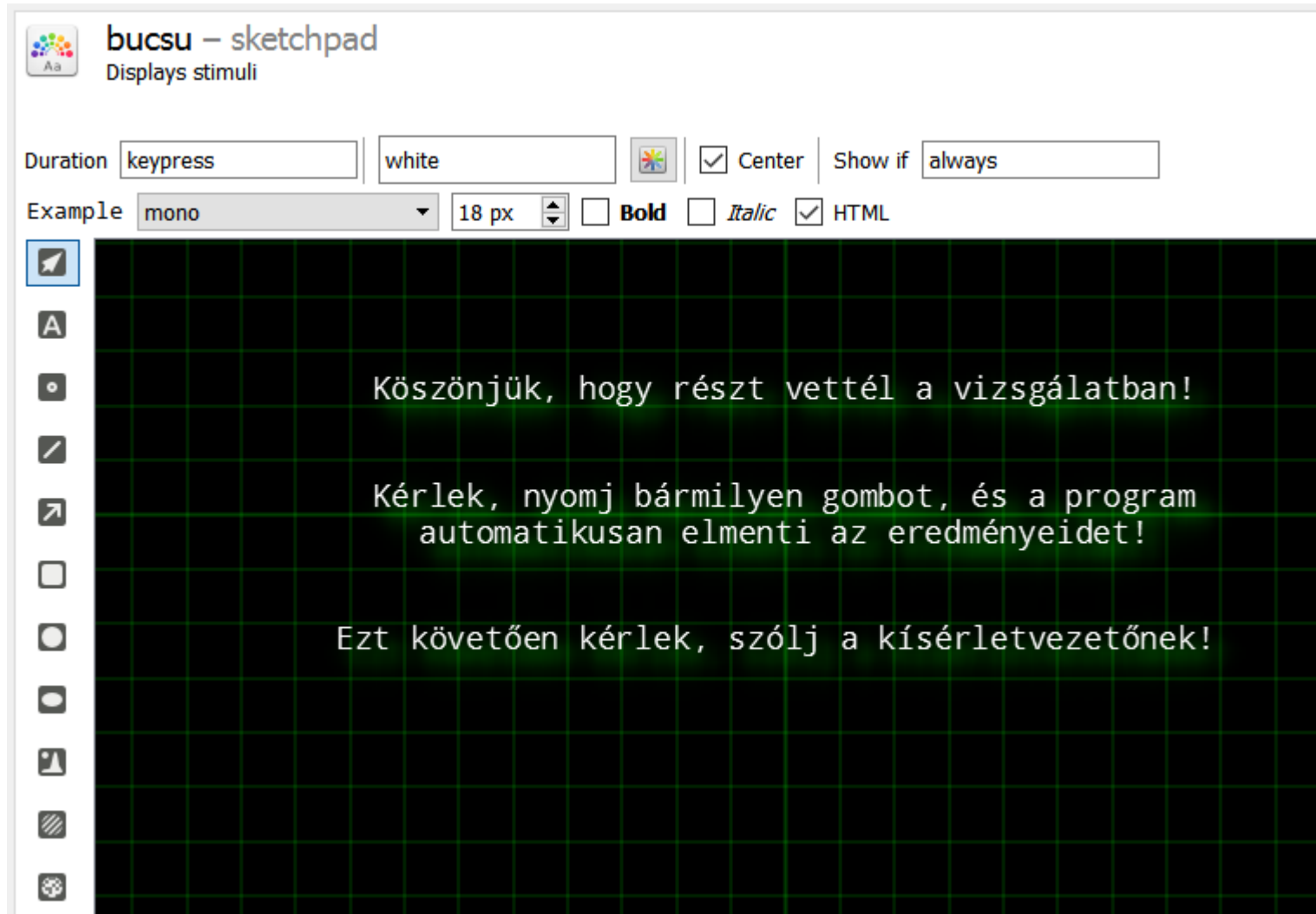
 **logger – logger**
Logs experimental data

☒ Log all variables (recommended)

 Add custom variable

	Custom variable	Source(s)
--	-----------------	-----------

- 10. búcsú
- Köszönd meg a ksz-nek a vizsgálatban való részvételt, és esetleg lásd el instrukciókkal, ezek után mit csináljon (pl. szóljon a kísérletvezetőnek)



Próbáljuk ki a programunkat!

- Próbáld ki a programot, közben figyeld, hogyan változnak az értékek a változó ablakban!

2. ezzel indítod a kísérletet 1. itt nyithatod meg a változó ablakot



