StimuliApp DOCUMENTATION

Source code

StimuliApp is an open source project. You can find the code in our GitHub repository.

Developed and maintained by

Brain circuits and behavior lab

Contact

marinraf@gmail.com

More info and tutorials

www.stimuliapp.com

Citations

In the following paper we present StimuliApp, its features, some implementation details and some assessment of timing performance. If you use StimuliApp you can cite it:

Marin-Campos, R., Dalmau, J., Compte, A., & Linares, D. (2020). StimuliApp: psychophysical tests on mobile devices. PsyArXiv. June 30. doi:10.31234/osf.io/yqd4c.

Other works:

Linares, D., Marin-Campos, R., Dalmau, J., & Compte, A. (2018). Validation of motion perception of briefly displayed images using a tablet. Scientific reports, 8(1), 1-6.

Linares, D., Amoretti, S., Marin-Campos, R., Sousa, A., Prades, L., Dalmau, J., Bernardo, M. & Compte, A. (2019). Perceptual spatial suppression and sensitivity for motion in schizophrenia. Schizophrenia Bulletin Open, in press.

iOS and macOS VERSIONS

v1.6.

Improvements:

5c5fe1a - ***Demonstration tests*** are included in the ***Tests*** menu to show the features of the app.

70b756b - A new variable (***scene_delayDisplay)*** is added to the results report.
The images to be presented on the screen are delivered with a certain constant frame rate.
From the moment the images are ***delivered*** until they are ***displayed*** on the screen, there is a small ***delay*** of about 10-30 ms depending on the hardware. This delay is approximately constant for each session although variable across sessions.
In previous versions, this delay was subtracted from the value of ***scene_responseTime.*** In this way, ***scene_responseTime*** measured the actual duration of the scene on the screen at the time of the response.
This correction, however, can produce ambiguities since all the other timers are synchronized with the ***delivery time*** rather than the ***display time***.
In this version, all timers are synchronized with the ***delivery time*** of the images and we added the variable ***scene_delayDisplay*** to the results report. In situations where the actual duration of the scene on the screen at the time of the response is needed (for example, reaction time experiments to visual stimulation), the user just need to subtract the value of ***scene_delayDisplay*** from the value of ***scene_responseTime***.

70b756b - A new response type (***lift finger)*** is included.
The response is triggered when the user lifts their finger from the screen.
This type of response can be useful, for example, to measure reaction times, combined with a previous touch response that triggers a stimulus.

Bug fixes:

54190a4 - Avoid using MTKTextureLoader when the code is running in the simulator, as the use of these textures is not supported and causes a crash.

46d1e36 - In the ***touch object response***, touching a non-interactive object now does not trigger the response of touching the background.

MAC VERSIONS

Beta 1.6.

- Beta version ported from the iOS version. Supporting all the features with no known bugs, but not fully tested yet.

NAVIGATION AND MENUS

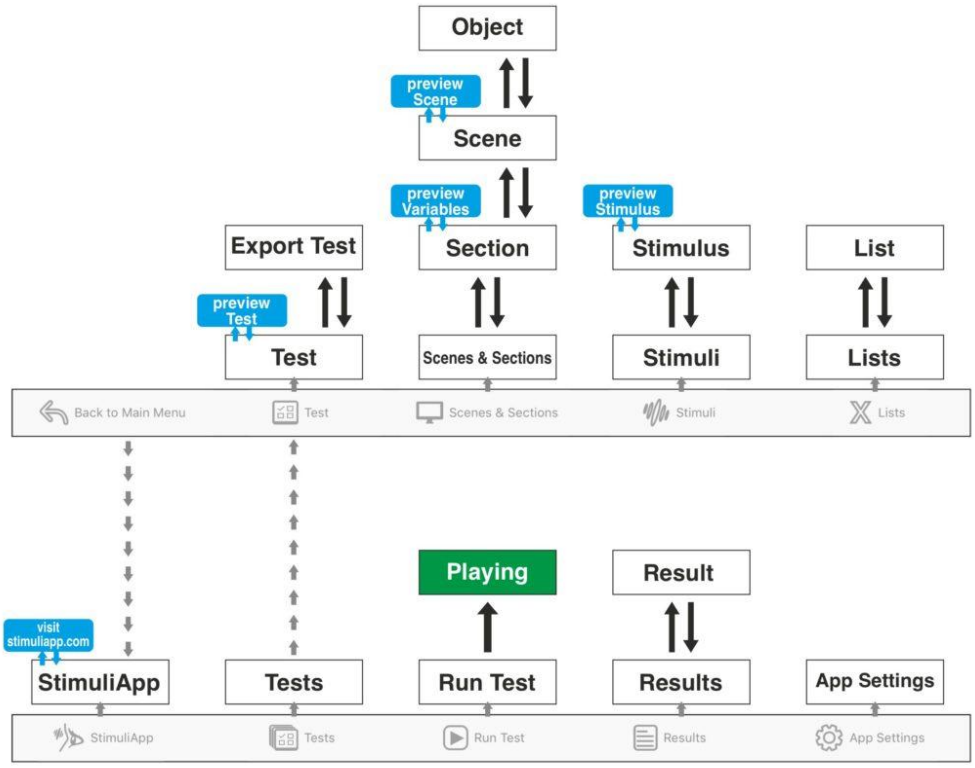
Navigation and menus

When you launch StimuliApp you see a navigation bar at the bottom of the screen with the following buttons:

- ***StimuliApp***: to go to the main menu.
- ***Tests***: to go to the menu to create, edit or delete tests.
- ***Run Test***: to run an existing test.
- ***Results***: to check the results of all the performed tests.
- ***App Settings***: to set and obtain information about the device.

When you start to edit a test, the navigation bar changes:

- ***Back to Main***: to return to the previous navigation bar.
- ***Test***: to access the general properties of the test.
- ***Scenes & Sections***: to manage the structure of the test. Tests are divided into sections. Each section contains one or more scenes containing objects.
- ***Stimuli***: to create stimuli templates that can be used in any scene in the test.
- ***Lists***: to create lists of numeric values, vectors, images, videos, audios, texts or blocks.



APP SETTINGS

User settings

- ***name***: A name to identify this device.
- ***email***: The default email address that will be used to send the results.

Device settings

- ***description***: The description of the device in use.
  If the device is not correctly identified, an alphaNumeric code preceded by the word “unknown” is displayed.
- ***system***: The OS system of the device.
- ***maximumFrameRate***: The maximum frame rate of the screen. This value can be 60 or 120 Hz depending on the device.
- ***maximumLuminance***: The maximum luminance of the device in cd/m².
  StimuliApp identifies most of the devices and sets as a default value for maximumLuminance the value included in the specifications by Apple. The nominal maximum luminance values were retrieved from apple.com and might slightly differ from the displayed values due to variations across series or the time in use of the displays.
  You can also manually specify maximumLuminance if you have measured the maximum luminance using a photometer.
  StimuliApp uses the maximumLuminance parameter, so you can directly read the luminance of your tests and stimuli directly in cd/m².
- ***audioRate***: The sampling frequency at which sounds are played.
- ***screenResolution***: Screen resolution in landscape mode.
- ***ppi***: The pixel density per inch of the screen.
  This value is provided directly when the device is identified. If the device is not correctly identified, you can provide a value manually.
  This value is only necessary if you plan to work in centimeters, inches or visual degrees instead of pixels.
- ***rampTime***: The transition time in seconds that any change in volume of the sounds requires.
  0.005 seconds is the recommended value to avoid pops and clicks that can occur when changes in volumes occur very quickly.
  From the startTime of a sound stimulus to startTime + rampTime, the volume changes from zero to the corresponding value.
  From startTime + duration – rampTime to startTime + duration the volume changes from the corresponding value to zero.

- ***delayAudio60***: Delay correction (negative or positive) to apply for the audiovisual synchronization on tests where the screen frame rate is 60Hz.
  When the user specifies that the auditory and visual signals should be presented at the same, some small audiovisual delays occur. The average delay is around -10 to 10 ms depending on the device. It is possible to correct this average delay.
  You need to present several times an audiovisual signal specified to be presented at the same time and calculate the average delay measuring the signals with an oscilloscope. Then, you should include the average delay in the delayAudio60 variable using a positive sign if you want that the correction delays the auditory signal presentation and a negative sign if you want that the correction delays the visual signal presentation.
  The variability of the delay across presentations (precision) is less than 1 millisecond (standard deviation) and cannot be corrected.
- ***delayAudio120***: Delay correction to apply for the audiovisual synchronization on tests where the screen frame rate is 120 Hz.
  This delay correction is necessary because the average delay is different when the device is working at 60 or 120 Hz.

Device settings (macOS version)

For the ***macOs*** version, the device settings are the same except for the ***screenResolution*** and ***ppi*** properties, which are replaced by:

- ***testWindowSize***: Tests are always run and previewed with StimuliApp in fullscreen mode.
  When a test is run, a test window is created with the size indicated in this property.
  Depending on your computer's CPU and GPU model, you may find that running tests using a large window results in a drop in performance and unstable frame rate.
  If that is the case, you can try reducing the size of the window.
  You can change the position of the window in the screen with the testWindowPosition properties.
- ***testWindowPositionX***: The x position of the test window, from the upper left corner of the screen.
- ***testWindowPositionY***: The y position of the test window, from the upper left corner of the screen.
- ***ppi***: The pixel density per inch of the screen. It is important that you calculate this value yourself, if you plan to work in centimeters, inches or visual degrees.
  Do not use the value provided by Apple or the manufacturer of your screen, because the value of your current ppi can depend on some system settings.
  The only way to be sure that you are using the right value for the ppi is to measure it yourself. To do that, first measure the size of the test window in pixels and then the length of the diagonal of the test window in inches. With these values you can easily get your ppi, for example using: https://www.pxcalc.com.
  Always verify that the ppi value is correct by drawing a rectangle of a certain size in cm or inches and measuring the actual size.

TEST PROPERTIES

Test name

- ***name***: A name to identify the test.

Test settings

- ***frameRate***: The desired frame rate of the screen. On some devices the refresh rate of the screen is always 60 Hz, in other devices it can be 60 or 120 Hz.

- ***luminance:*** You can control the luminance of the screen with this parameter.
  The perceived brightness is approximately proportional to the logarithm of the luminance that you set with this parameter.
  The new luminance will only be effective once the test begins.
  There are a few preferences on iOS and iPadOS that can automatically change the brightness and color temperature settings of the device:
  Auto-Brightness can be found in your device Settings, Accessibility, Display & Text Size.
  TrueTone and ***Night Shift*** options can be found in your device Settings, Display & Brightness.
  Remember to disable these options to avoid unwanted changes of brightness when running a test.
  Even with the Auto-Brightness adjustment disabled, the brightness of the device can be slightly increased automatically if you are under a bright light (outside).

- ***viewingDistance***: Viewing distance from the participant to the screen. This value is used to calculate the actual pixel size of any property that is measured in visual angle degrees.
  - ***constant***: The distance from the participant to the screen is a constant value.
  - ***set each time***: Each time you run the test, you are asked to enter the distance from the participant to the screen.

- ***XButton***: The position on the screen of the button that is used to cancel a test in progress.

- ***randomness***: How random numbers are generated.
  - automatically generated: The random numbers are generated automatically each time you run the test.
  - generated with seeds: Each time you run the test, you are asked for numeric seeds to generate the random numbers.

- ***gamma***: Establish how the luminance raises with the input value.
  - linear: The gamma is transformed to make the luminance linear, assuming the screen has a gamma value of 2.2, which is the gamma used in iOS and iPadOs devices. The correction is simply to raise the luminance to the power of 1/2.2
  - normal: No transformation is made. The values are not in linear space. This option is a good choice if you are drawing images.
  - calibrated: If you are using stimuliApp on a computer or with an external monitor, you may want to calibrate it with a photometer and manually enter the correction value for gamma. The correction is simply to raise the luminance to the power of 1/gamma.

First section

- ***firstSection***: The first section of the test.

SECTION PROPERTIES

Section name

- ***name***: A name to identify this section.

Scenes

All the scenes in this menu will be presented consecutively, one after the other, in the order they are in.
The presentation of all the scenes in a section is called a ***trial***.

Variables in the section

Variables of all the objects in any scene in the section.
We call variables the properties of an object that are not kept constant across trials, but vary from trial to trial.
Select any of the variables to manage what their possible values are and how they are chosen in each trial.
The variables are always named: ***sceneName_objectName_propertyName***.

Repetitions and trials

- ***repetitions***: The number of repetitions of all the different trials.
- ***numberOfDifferentTrials***: The number of different ways in which the possible values of the variables can be assigned.
- ***totalNumberOfTrials***: numberOfDifferentTrials * repetitions.

Trial Value

- ***trialValueVariable***: It is possible to associate a different value for each trial in the section. To do this, it is necessary to select the variable that will be used to calculate the value of each trial. If no variable is selected, the value of each trial will always be considered zero.
    - ***same***: The value of the trial is equal to the value of the variable that we have selected. It can be a numeric value or a position vector.
    - ***other***: The value of the trial is equal to a numeric value set for each of the possible values of the variable that we have selected.

Response Value

- ***responseValueParameter***: It is possible to select one of the parameters of any response to be the responseValue of the section. The responseValue can be a numeric value or a position vector.
    - ***marginError***: We compare ***responseValue*** with ***trialValue*** and calculate their difference. If they are numeric values:***difference = abs(responseValue – trialValue)***. If they are vectors, the difference is the distance between the points they represent on the screen. If ***distance < marginError*** the trial is considered ***correct***, otherwise it is considered ***incorrect***. The variable used to calculate the trialValue and the response parameter used to calculate the responseValue must have the same units for the comparison to be fair. For example, if the trialValue is determined by the anglePosition of a certain stimulus and the responseValue is determined by the anglePosition of the touch on the screen, both angle positions must be measured in radians or both must be measured in degrees.

End of trial conditions

- ***End of trial conditions***
    - ***conditions***: It is possible to create conditions that are evaluated after each trial. The order of the conditions in the menu is the order in which they are evaluated. If a condition is true, the action associated with that condition will be performed and the following conditions will not be evaluated. The possible actions that can be performed are to go to another section or to end the test. If neither condition is true, a new trial in the same section is performed.
    - ***when all trials have been performed***: This condition is always the last of all possible conditions and it is true only when all possible trials of the section have been performed.

SCENE PROPERTIES

Scene name

- ***name***: A name to identify the scene.

Scene duration

- ***duration***: The maximum duration of the scene if no response is given.
    - ***constant***: The duration of the scene is a fixed value.
    - ***endOfStimuli***: The scene ends when all stimuli end.

Scene response

- ***response***: To select one of the possible types of responses for the scene. Each time a response is given the scene ends even if the time has not reached the duration of the scene.

Background & layers

- ***color***: Color of the background.

- ***numberOfLayers***: The number of layers in the scene.
  By default, all objects are drawn on the same layer, which is drawn on top of the background. If you want to draw one object on top of another, you must increase the number of layers. (This does not apply to video or text objects that are rendered differently, always on top of everything).
  When drawing objects, a preassigned space is saved for each one of them. This space is slightly larger than the object itself. If two objects are too close to each other, their respective preassigned spaces can interfere, creating an unwanted empty space between the objects. To avoid this, you should increase the number of layers (or separate the objects a bit if possible).
  Increasing the number of layers is computationally costly, so try to use as few layers as possible, especially when working at 120Hz.

- ***continuosResolution***: When using stimuliApp, images are displayed in an sRGB space with 256 possible values for each channel of the RGB color. If you are working only with luminances: R = G = B, there are 256 luminance levels.
  Sometimes this limitation on the number of different luminance intensities displayable can be an issue.
  By making the continuosResolution property true, the noisy-bit method is implemented.
  This method consists of adding a small amount of random noise. The method is described in *Allard, R., Faubert, J., 2008. The Noisy-Bit method for digital displays: Converting a 256 luminance resolution into a continuous resolution. Behav. Res. Methods 40, 735–743*.
  The noisy-bit method, combined with the 256 luminance levels, is perceptually equivalent to an analog display with a continuous luminance intensity resolution when the spatiotemporal resolution is high enough that the noise becomes negligible.

Objects

- ***background***: Background object that sets the color of the screen.

- ***objects***: Objects are drawn on the screen in the same order as they are in this menu, except for video or text objects that are always rendered above everything else. You can change the order of the objects by clicking the Edit button and moving them in the menu.

STIMULUS PROPERTIES

Stimulus name

- ***name***: A name to identify the stimulus.

Stimulus type

patch

Patch with uniform color.

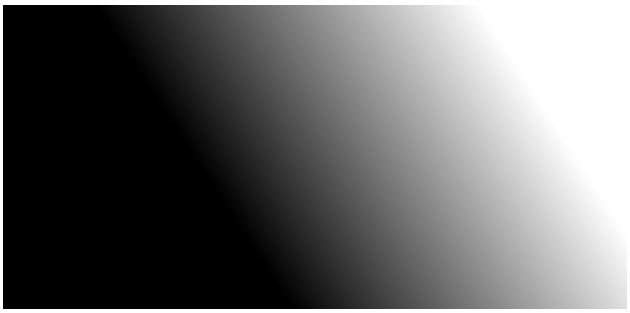
patch properties

- ***color***: Color of the patch. Color property. Time-dependent.

other properties

- ***Shape and size***
- ***Duration***
- ***Position***
- ***Border***
- ***Contrast***
- ***Noise***
- ***Modulator of contrast***

gradient



Patch with gradient color. Changing linearly from color1 to color2 for the size of the gradient.

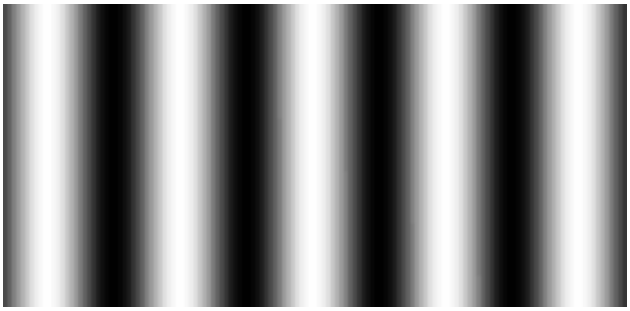
gradient properties

- ***gradientSize***: The size of the gradient. It is defined as the distance from the position with color = color1 to the position with color = color2. Size property. Time-dependent.
- ***color1***: First color. Color property. Time-dependent.
- ***color2***: Second color. Color property. Time-dependent.
- ***gradientPosition***: Position of the center of the gradient relative to the center of the shape that contains it. Measured in the direction of the gradient set by the gradientRotation. Position property. Time-dependent.
- ***gradientRotation***: The orientation of the gradient. This property only rotates the gradient but not the shape that masks it. Angle property. Time-dependent.

other properties

- ***Shape and size***
- ***Duration***
- ***Position***
- ***Border***
- ***Contrast***
- ***Noise***
- ***Modulator of contrast***

grating



Grating with color sinusoidally oscillating from color1 to color2.

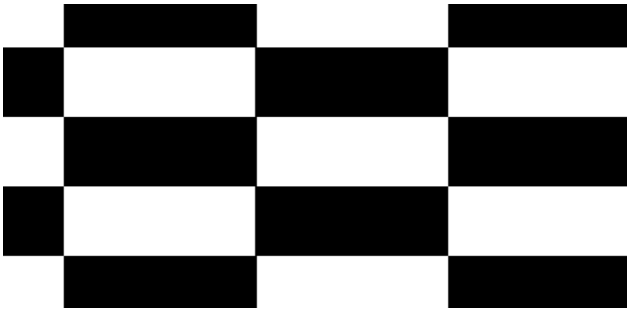
grating properties

- ***period***: Size period of the sinusoidal function. Size property. Time-dependent.
- ***color1***: First color. Color property. Time-dependent.
- ***color2***: Second color. Color property. Time-dependent.
- ***phase***: Phase of the sinusoidal function. Position property. Time-dependent.
- ***gratingRotation***: Orientation of the grating. This property only rotates the grating but not the shape that masks it. Angle property. Time-dependent.

other properties

- ***Shape and size***
- ***Duration***
- ***Position***
- ***Border***
- ***Contrast***
- ***Noise***
- ***Modulator of contrast***

checkerboard



Checkerboard defined by two different colors and the size of the boxes.

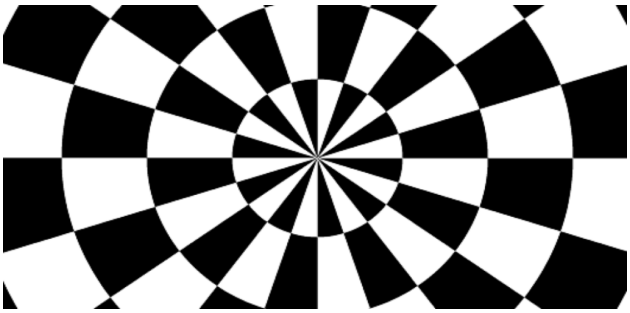
checkerboard properties

- ***boxSize***: Horizontal and vertical size of the boxes. 2dSize property. Time-dependent.
- ***color1***: Color of the first box. Color property. Time-dependent.
- ***color2***: Color of the second box. Color property. Time-dependent.
- ***checkerboardPosition***: Position of the center of the checkerboard relative to the center of the shape that contains it. 2dPosition property. Time-dependent.
- ***checkerboardRotation***: Orientation of the checkerboard. This property only rotates the checkerboard but not the shape that masks it. Angle property. Time-dependent.

other properties

- ***Shape and size***
- ***Duration***
- ***Position***
- ***Border***
- ***Contrast***
- ***Noise***
- ***Modulator of contrast***

radialCheckerboard



Checkerboard with radial symmetry defined by two different colors and the angle size of the boxes. Divided into different circles (maximum 10 divisions).

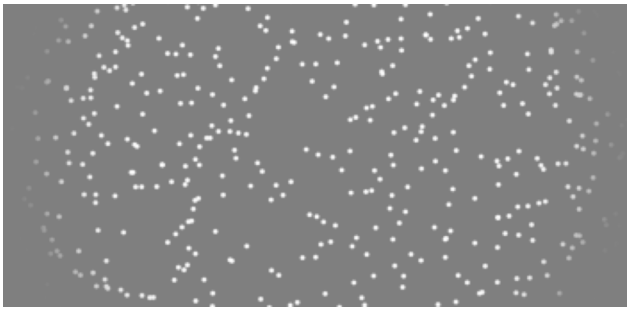
radialCheckerboard properties

- ***boxAngleSize***: Angle size of the boxes. Angle property. Time-dependent.
- ***color1***: Color of the first box. Color property. Time-dependent.
- ***color2***: Color of the second box. Color property. Time-dependent.
- ***checkboardRotation***: Orientation of the checkerboard. This property only rotates the checkerboard but not the shape that masks it. Angle property. Time-dependent.
- ***diameter1***: Diameter of the first circle. Size property. Time-dependent.
- ***diameter2***: Diameter of the second circle. Size property. Time-dependent.
- ***diameter3***: Diameter of the third circle. Size property.Time-dependent.
- ***diameter4***: Diameter of the fourth circle. Size property. Time-dependent.
- ***diameter5***: Diameter of the fifth circle. Size property. Time-dependent.
- ***diameter6***: Diameter of the sixth circle. Size property. Time-dependent.
- ***diameter7***: Diameter of the seventh circle. Size property. Time-dependent.
- ***diameter8***: Diameter of the eighth circle. Size property. Time-dependent.
- ***diameter9***: Diameter of the ninth circle. Size property. Time-dependent.
- ***diameter10***: Diameter of the tenth circle. Size property. Time-dependent.

other properties

- ***Shape and size***
- ***Duration***
- ***Position***
- ***Border***
- ***Contrast***
- ***Noise***
- ***Modulator of contrast***

dots



Random dots.

dots properties

- ***density***: The density of dots per pixel. Value from zero to one.
- ***coherence***: The proportion of type1 dots from the total number of dots. Value from zero to one. Time-dependent.
- ***dotsLife***: The life of each one of the dots. When a dot reaches their life it disappears and another dot is created in a random position. Time property.
- ***behaviour***: Possible types of behaviour.
    - ***same***: The dot is always the same type (type1 or type2) during all its life.
    - ***different***: Each frame a dot can change from type1 to type2 or vice versa.
- ***diameter1***: The diameter of type1 dots. Size property. Time-dependent.
- ***direction1***: Possibles ways to establish the dot direction for type1 dots.
    - ***random***: The direction of movement of the dots is random.
    - ***fixed***: The dots move in a fixed direction.
    - ***center***: The dots move towards the stimulus center.
    - ***away from the center***: The dots move away from the stimulus center.
    - ***clockwise***: The dots move clockwise.
    - ***counterclockwise***: The dots move counterclockwise.
- ***color1***: The color of type1 dots. Color property. Time-dependent.
- ***diameter2***: The diameter of type2 dots. Size property. Time-dependent.
- ***direction2***: Possibles ways to establish the dot direction for type2 dots.random: The direction of movement of the dots is random.
    - ***fixed***: The dots move in a fixed direction.
    - ***center***: The dots move towards the stimulus center.
    - ***away from the center***: The dots move away from the stimulus center.
    - ***clockwise***: The dots move clockwise.
    - ***counterclockwise***: The dots move counterclockwise.
- ***color2***: The color of type2 dots. Color property. Time-dependent.

other properties

- ***Shape and size***
- ***Duration***
- ***Position***
- ***Border***
- ***Contrast***
- ***Noise***
- ***Modulator of contrast***

image

Image.

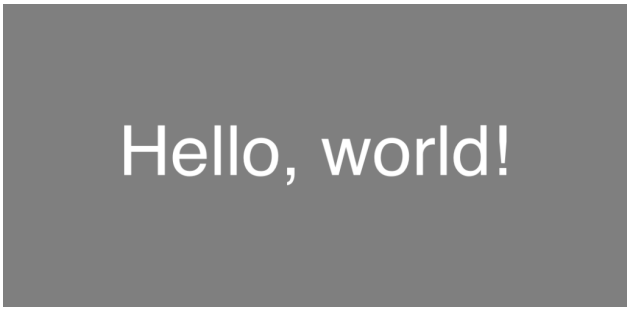
image properties

- ***imageNumber***: The number of the image selected from the list of images. Positive non-zero integer.
- ***imagePosition***: Position of the center of the image relative to the center of the shape that contains it. 2dPosition property. Time-dependent.
- ***imageRotation***: Orientation of the image. This property only rotates the image but not the shape that masks it. Angle property. Time-dependent.

other properties

- ***Shape and size***
- ***Duration***
- ***Position***
- ***Border***
- ***Contrast***
- ***Noise***
- ***Modulator of contrast***

text



Text.

text properties

- ***textNumber***: The number of the text selected from the list of texts. Positive non-zero integer.
- ***font***: The font of the text. Font property.
- ***textSize***: Size of the text in points. Positive non-zero integer.
- ***textPositionX***: The x position relative to the center of the screen. Size property.
- ***textPosition***: The y position relative to the center of the screen. Size property.
- ***color***: Color of the text. Color property.

other properties

- ***Duration***

video

Play video from a source.

video properties

- ***videoNumber***: The number of the video selected from the list of videos. Positive non-zero integer.

other properties

- ***Duration***

audio

Play audio from a source.

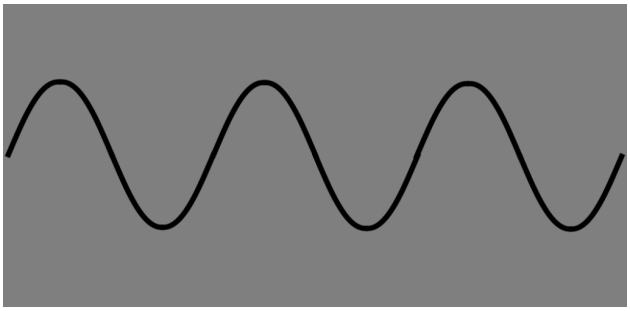
audio properties

- ***audioNumber***: The number of the audio selected from the list of audios. Positive non-zero integer.

other properties

- ***Duration***

pureTone



Play auto-generated white noise audio or pure tone audio.

pureTone properties

- ***soundType***: Select between pure tone sound (tone with a simple sinusoidal waveform) or white noise sound (random signal having equal intensity at different frequencies).
- ***frequency***: The frequency of the sine wave (Only in the case of pure tone sound). Frequency property.
- ***amplitude***: The amplitude of the sine wave. The perceived loudness of sound is basically proportional to the logarithm of the amplitude of the sine wave. Value from zero to one.
- ***leftRightBalance***: From 0 = totally to the left to 1 = totally to the right. Value from zero to one.
    - ***leftAmplitude*** = leftRightBalance * amplitude
    - ***rightAmplitude*** = (1 – leftRightBalance) * amplitude

other properties

- ***Duration***

Shape and size

rectangle

Rectangle or square shape.

rectangle properties

- ***size***: The size of the shape containing the stimulus. 2dSize property. Time-dependent.

ellipse

Ellipse or circle shape.

ellipse properties

- ***size***: The size of the shape containing the stimulus. 2dSize property. Time-dependent.

cross

Cross shape.

cross properties

- ***length***: The length of the cross sides. Size property. Time-dependent.
- ***thickness***: The thickness of the cross strokes. Size property. Time-dependent.

polygon

Polygon (3 to 10 sides) shape.

polygon properties

- ***diameterSize***: The diameter of the circumscribed circle of the polygon. Size property. Time-dependent.
- ***sides***: The number of sides of the polygon. Integer from 3 to 10. Time-dependent.

ring

Ring shape.

polygon properties

- ***exteriorDiameter***: The exterior diameter of the ring. Size property. Time-dependent.
- ***interiorDiameter***: The interior diameter of the ring. Size property. Time-dependent.

wedge

Wedge shape.

wedge properties

- ***diameter***: The diameter of the circle the wedge is part of. Size property. Time-dependent.
- ***angleSize***: The angle size of the wedge. Angle property. Time-dependent.

Duration

- ***activated***: Boolean variable that establishes whether the stimulus is active or not.
  Useful when you have stimuli that should be presented in some trials and not presented in other trials.
  Use this variable to control the presence of the stimulus rather than using contrast = 0 or volume = 0 or other workarounds.
  Making the stimulus inactive is the only way to make sure neither CPU or GPU cycles are used to compute the stimulus. Integer value 0 or 1.
- ***start***: Time at which the stimulus begins, measured from the start of the scene. Time property.
- ***duration***: Duration of the stimulus. By default is set to 1000 seconds. Time property.

Position

- ***originCoordinates***: By default, the origin of coordinates is placed in the center of the screen, but it is possible to place it in another point of the screen. 2dPosition property. Time-dependent.
- ***position***: Position of the center of the stimulus relative to the origin of coordinates. X increases from left to right and Y increases from bottom to top. The angles are measured counterclockwise from the X-axis. 2dPosition property. Time-dependent.
- ***rotation***: The angle of rotation of the stimulus (both the shape and its content). Measured counterclockwise. Angle property. Time-dependent.

Border

- ***border***: Border around the stimulus.
  - ***none***: No border.
  - ***normal***: The border has the same contrast, noise and modulation as the stimulus.
    - ***borderDistance***: Distance from the interior limit of the border to the exterior limit of the stimulus. The distance can be positive or negative to create the border in the exterior or interior of the stimulus. Size property. Time-dependent.
    - ***borderThickness***: The size of the border. Size property. Time-dependent.
    - ***borderColor***: The color of the border. Color property. Time-dependent.
  - ***opaque***: The border has always contrast = 1 and zero noise and modulation, independently of the stimulus values.
    - ***borderDistance***: Distance from the interior limit of the border to the exterior limit of the stimulus. The distance can be positive or negative to create the border in the exterior or interior of the stimulus. Size property. Time-dependent.
    - ***borderThickness***: The size of the border. Size property. Time-dependent.
    - ***borderColor***: The color of the border. Color property. Time-dependent

Contrast

- ***contrast***: Contrast of the stimulus.
  - ***uniform***: The contrast is uniform.
    - contrastValue: Numeric value of the contrast. Value from zero to one. Time-dependent.
  - ***gaussian***: The contrast is fitted by a 2d gaussian function centered in the center of the shape.
    - contrastValue: Maximum value of the contrast in the center of the stimulus. Value from zero to one. Time-dependent.
    - contrastGaussianDeviation: The standard deviation of the gaussian function of the contrast. Size property. Time-dependent.
  - ***cosine***: The contrast is fitted by a 2d raised cosine function centered in the center of the shape.
    - contrastValue: Maximum value of the contrast in the center of the stimulus. Value from zero to one. Time-dependent.
    - contrastCosineValue: The stimulus is masked with a circular shape of diameter equal to the maximum size of the stimulus. In the exterior part of the shape, the contrast decay from contrastValue to zero following a cosine function. ContrastCosineValue is the proportion of the shape that is not affected by the cosine decay. Value from zero to one. Time-dependent.

Noise

- ***none***: No noise.

- ***gaussian***: Gaussian noise. Adding a noise RGB value (x;x;x) to the color. The x value is a random value calculated from a normal distribution with mean = 0 and deviation = noiseDeviation.
  - ***noiseDeviation***: The value of the standard deviation of the normal distribution that adds noise to the stimulus. Positive (or zero) decimal value. Time-dependent.
  - ***noiseTimePeriod***: The duration of each different noise calculation. If you set this value to 1 frame the noise will change each frame. If you want a static noise, you can make this value longer than the duration of the stimulus. Time property.
  - ***noisePosition***: The position of the origin of the noise. Making this parameter dependent on time you can move the noise in a certain direction. 2dPosition property. Time-dependent.
  - ***noiseRotation***: The rotation of the noise. Angle property. Time-dependent.
  - ***noiseSize***: The size of the blocks with the same noise. From 1 pixel to the size of the stimulus. 2dSize property. Time-dependent.

- ***perlin***: Simple Perlin noise implementation with a parameter for the intensity of noise and another parameter for smoothness.
  - ***noiseIntensity***: Intensity of Perlin noise that is added to the stimulus. Value from zero to one. Time-dependent.
  - ***noiseTimePeriod***: The duration of each different noise calculation. If you set this value to 1 frame the noise will change each frame. If you want a static noise, you can make this value longer than the duration of the stimulus. Time property.
  - ***noisePosition***: The position of the origin of the noise. Making this parameter dependent on time you can move the noise in a certain direction. 2dPosition property. Time-dependent.
  - ***noiseRotation***: The rotation of the noise. Angle property. Time-dependent.
  - ***noiseSmoothness***: The smoothness value of the Perlin noise implementation. Value from zero to one. Time-dependent.

Modulator of contrast

- ***modulator***: It is possible to modulate the contrast of a stimulus to create second-order patterns. A common example would be a random noise texture modulated by the contrast of a larger-scale sinusoidal pattern.
  - ***none***: No modulator of contrast.
  - ***sinusoidal***: The contrast is modulated by a sinusoidal function.
    - ***modulatorAmplitude***: The amplitude of the sine modulator of the contrast. Value from zero to one. Time-dependent.
    - ***modulatorPeriod***: The size period of the sine modulator of the contrast. Size property. Time-dependent.
    - ***modulatorPhase***: The phase of the sine modulator of the contrast. Angle property. Time-dependent.
    - ***modulatorRotation***: The rotation of the sine modulator of the contrast. Angle property. Time-dependent.

LISTS

You can create different types of lists:

List of numeric values

The values in the list do not have any unit. A numeric value of 3 can mean: 3 cm, 3 inches, 3 seconds, 3 frames, 3 radians… depending on the unit of the variable that acquires its values from the list.

- ***name***: A name to identify the list.
- ***shuffled***: If the values in the list are always in the same order or if they are shuffled every time the test is run.
    - ***in order***: The values in the list are always in the same order. When an object selects values from the list, it can select the values in order or randomly, but the list itself is always in the same order.
    - ***shuffled***: The values in the list are shuffled once at the beginning of each test.
- ***jittering of the values***: If the jittering is not zero, a numeric value in the range (-jitteringValue, jitteringValue) is randomly generated in each trial and added to the corresponding value from the list.

List of 2d vectors (size or position)

The values in the list do not have any unit. A value of (2;6) can mean: 2 cm; 6cm if the variable that acquires its values from the list is cartesian and the units are cm. But it can also mean: 2 inches and 6 degrees, for example, if the variable is polar and measured in those units.

- ***name***: A name to identify the list.
- ***shuffled***: If the values in the list are always in the same order or if they are shuffled every time the test is run.
    - ***in order***: The values in the list are always in the same order. When an object selects values from the list, it can select the values in order or randomly, but the list itself is always in the same order.
    - ***shuffled***: The values in the list are shuffled once at the beginning of each test.

List of 3d vectors (color)

The values of the list measure color as (R;G;B).

- ***name***: A name to identify the list.
- ***shuffled***: If the values in the list are always in the same order or if they are shuffled every time the test is run.
    - ***in order***: The values in the list are always in the same order. When an object selects values from the list, it can select the values in order or randomly, but the list itself is always in the same order.
    - ***shuffled***: The values in the list are shuffled once at the beginning of each test.

List of Images

To draw images in your test, you must first create a list of images. In that list you must add all the images you want to use.
Once inside the list of images, press +add image and the media library will open. Select an image from the library and it will be imported to your test. You can use any image format supported by the media library.
If you try to add an image and it is not imported, try using a .jpg or .png file instead.

- ***shuffled***: If the values in the list are always in the same order or if they are shuffled every time the test is run.
    - ***in order***: The values in the list are always in the same order. When an object selects values from the list, it can select the values in order or randomly, but the list itself is always in the same order.
    - ***shuffled***: The values in the list are shuffled once at the beginning of each test.

List of Texts

To draw texts in your test, you must first create a list of texts. In that list, you must add all the texts you want to use.
Once inside the list of texts, press +add text and use the keyboard to write the text.

- ***shuffled***: If the values in the list are always in the same order or if they are shuffled every time the test is run.
    - ***in order***: The values in the list are always in the same order. When an object selects values from the list, it can select the values in order or randomly, but the list itself is always in the same order.
    - ***shuffled***: The values in the list are shuffled once at the beginning of each test.

List of Videos

To play videos in your test, you must first create a list of videos. In that list you must add all the videos you want to use.
Once inside the list of videos, press +add video and the media library will open. Select a video from the library and it will be imported to your test. You can use any video format supported by the media library.
If you try to add an image and it is not imported, try using a .mov or .mp4 file instead.

- ***shuffled***: If the values in the list are always in the same order or if they are shuffled every time the test is run.
    - ***in order***: The values in the list are always in the same order. When an object selects values from the list, it can select the values in order or randomly, but the list itself is always in the same order.
    - ***shuffled***: The values in the list are shuffled once at the beginning of each test.

List of Audios

To play audios in your test, you must first create a list of audios. In that list you must add all the audios you want to use.
Once inside the list of audios, press +add audio and the music library will open. Select any audio from the library and it will be imported to your test. You can use any audio format supported by the music library. You will not be able to import files with DRM restrictions.

- ***shuffled***: If the values in the list are always in the same order or if they are shuffled every time the test is run.
    - ***in order***: The values in the list are always in the same order. When an object selects values from the list, it can select the values in order or randomly, but the list itself is always in the same order.
    - ***shuffled***: The values in the list are shuffled once at the beginning of each test.

Lists of Blocks

Blocks are more complex ways of organising the possible values of a variable. Visit Selection of variables for some usage examples.

Repetitions

- ***numberOfBlocks***: The number of blocks.
- ***lengthOfBlocks***: The length of the blocks (the number of trials for each one of the blocks).

Types of Block

- ***typesOfBlocks***: All blocks can be of the same type or there can be 2 different types of blocks.
- ***startingBlock***: The starting block of the test. (If the typesOfBlock is 2).
    - ***random***
    - ***first***
    - ***second***
- ***probChangeBlock***: The probability of changing the type of block when a block is finished. (If the typesOfBlock is 2).

First Block

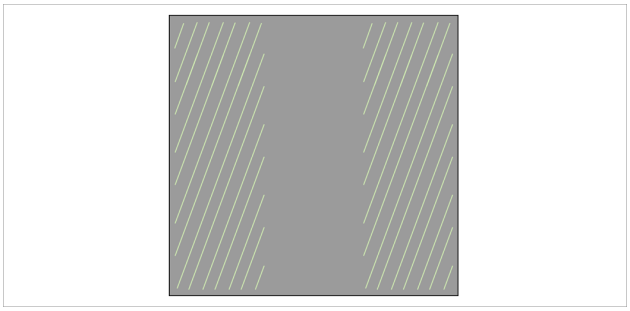
- ***firstBlockFirstList***: The first list of values for the first block.
- ***firstBlockSecondList***: The second list of values for the first block.
- ***firstBlockStartingList***: The starting list for the first block.
    - ***random***
    - ***first***
    - ***second***
- ***firstBlockProbChangeList***: The probability of changing the list when a trial of the first block ends.

Second Block

- ***secondBlockFirstList***: The first list of values for the second block.
- ***secondBlockSecondList***: The second list of values for the second block.
- ***secondBlockStartingList***: The starting list for the second block.
    - ***random***
    - ***first***
    - ***second***
- ***secondBlockProbChangeList***: The probability of changing the list when a trial of the second block ends.

RESPONSES

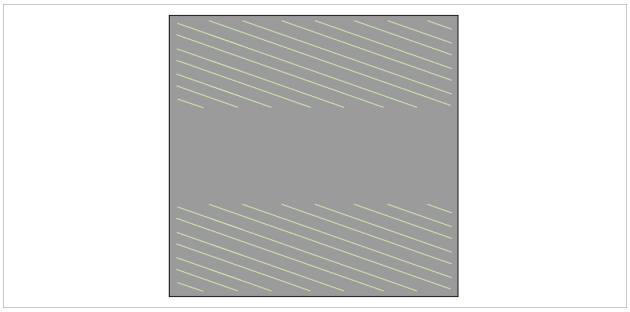
left or right



Any touch on the 1/3 left part of the screen is considered a left response.
Any touch on the 1/3 right part of the screen is considered a right response.
Any touch on the 1/3 central part of the screen is ignored.

- ***startTime***: Time from which it is possible to respond. By default it is zero, the user can respond at any time. Time property.
- ***endTime***: Maximum time until which it is possible to respond. Time property.
- ***wrongTiming***: If 0 it is not possible to respond before startTime or after endTime.
  If 1 it is possible to respond before startTime or after endTime but the response is considered incorrect. Integer value 0 or 1.
- ***leftValue***: Numeric value for the left response. Decimal value.
- ***rightValue***: Numeric value for the right response. Decimal value.

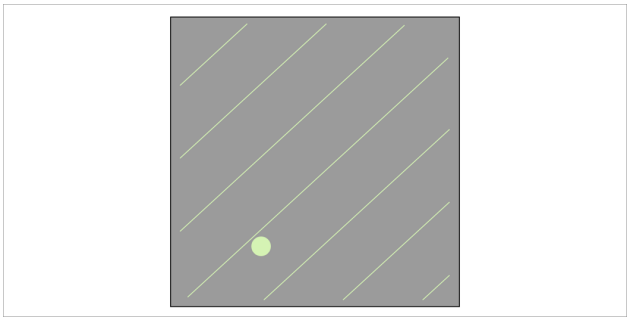
top or bottom



Any touch on the 1/3 top part of the screen is considered a top response.
Any touch on the 1/3 bottom part of the screen is considered a bottom response.
Any touch on the 1/3 central part of the screen is ignored.

- ***startTime***: Time from which it is possible to respond. By default it is zero, the user can respond at any time. Time property.
- ***endTime***: Maximum time until which it is possible to respond. Time property.
- ***wrongTiming***: If 0 it is not possible to respond before startTime or after endTime. If 1 it is possible to respond before startTime or after endTime but the response is considered incorrect. Integer value 0 or 1.
- ***topValue***: Numeric value for the top response. Decimal value.
- ***bottomValue***: Numeric value for the bottom response. Decimal value.

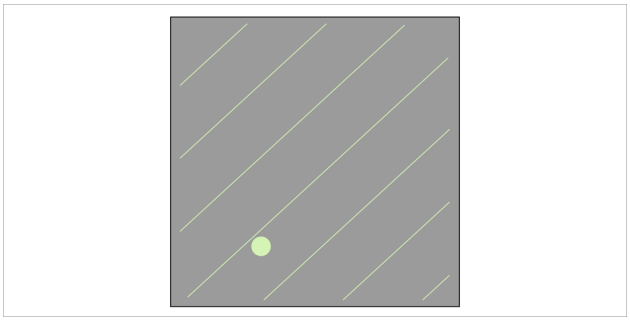
touch screen



Any touch on the screen is considered a response and the position (x, y) or (radius, angle) is saved.

- ***startTime***: Time from which it is possible to respond. By default it is zero, the user can respond at any time. Time property.
- ***endTime***: Maximum time until which it is possible to respond. Time property.
- ***wrongTiming***: If 0 it is not possible to respond before startTime or after endTime. If 1 it is possible to respond before startTime or after endTime but the response is considered incorrect. Integer value 0 or 1.
- ***originCoordinates***: By default, the center of the screen is the origin of coordinates but you can change the origin so that it is in any other point. 2dPosition property.
- ***position***: The response position, measured in cartesian or polar variables. 2dPosition property.

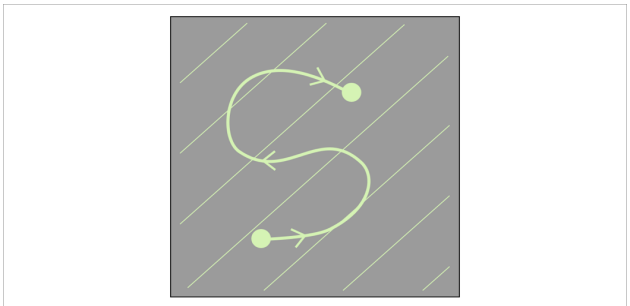
lift finger



The response is triggered when the user lifts their finger from the screen.
This type of response can be useful to measure reaction times, combined with a previous touch response that triggers a stimulus, for example.

- ***startTime***: Time from which it is possible to respond. By default it is zero, the user can respond at any time. Time property.
- ***endTime***: Maximum time until which it is possible to respond. Time property.
- ***wrongTiming***: If 0 it is not possible to respond before startTime or after endTime.
  If 1 it is possible to respond before startTime or after endTime but the response is considered incorrect. Integer value 0 or 1.
- ***liftValue***: Numeric value when the user lifts the finger on time. Decimal value.

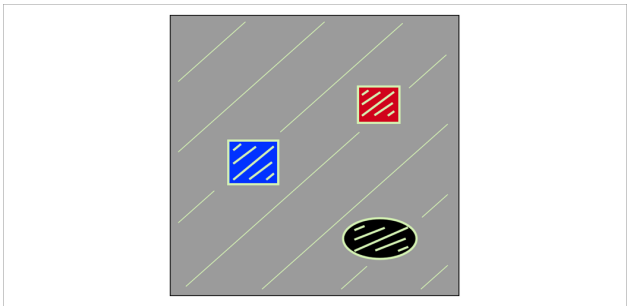
path



Any touch on the screen initiates a path that ends when the finger is lifted from the screen.
The path is saved as (x, y) or (radius, angle).

- ***startTime***: Time from which it is possible to respond. By default it is zero, the user can respond at any time. Time property.
- ***endTime***: Maximum time until which it is possible to respond. Time property.
- ***wrongTiming***: If 0 it is not possible to respond before startTime or after endTime. If 1 it is possible to respond before startTime or after endTime but the response is considered incorrect. Integer value 0 or 1.
- ***originCoordinates***: By default, the center of the screen is the origin of coordinates but you can change the origin so that it is in any other point. 2dPosition property.
- ***position***: The response position, measured in cartesian or polar variables. 2dPosition property.

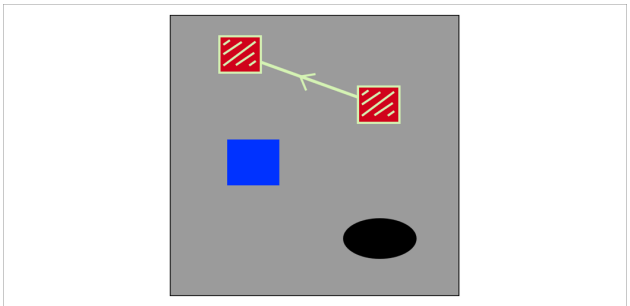
touch object



Touching certain objects on the screen can be considered a response with a certain numeric value associated.

- ***startTime***: Time from which it is possible to respond. By default it is zero, the user can respond at any time. Time property.
- ***endTime***: Maximum time until which it is possible to respond. Time property.
- ***wrongTiming***: If 0 it is not possible to respond before startTime or after endTime.
  If 1 it is possible to respond before startTime or after endTime but the response is considered incorrect. Integer value 0 or 1.
- ***object***: There will be one of these properties for each touchable object in the scene. To establish whether the object is interactive (can be touched) or not.
  - ***objectValue***: The value associated with the object. Decimal value.

move object



Move an existing object by touching it.
The path is saved as (x, y) or (radius, angle).

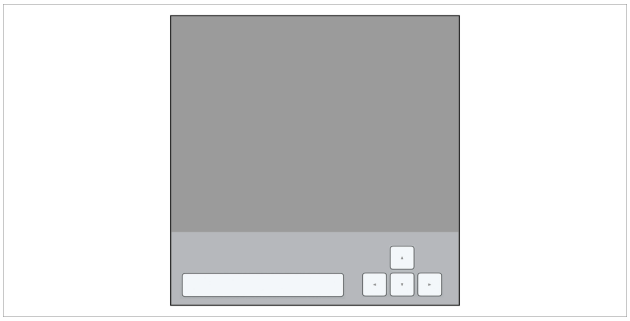
- ***startTime***: Time from which it is possible to respond. By default it is zero, the user can respond at any time. Time property.
- ***endTime***: Maximum time until which it is possible to respond. Time property.
- ***wrongTiming***: If 0 it is not possible to respond before startTime or after endTime. If 1 it is possible to respond before startTime or after endTime but the response is considered incorrect. Integer value 0 or 1.
- ***originCoordinates***: By default, the center of the screen is the origin of coordinates but you can change the origin so that it is in any other point. 2dPosition property.
- ***position***: The response position, measured in cartesian or polar variables. 2dPosition property.
- ***endPath***:
  - ***Lift***: the response path ends when the participant lifts their finger from the screen.
  - ***Touch object***: the response path ends when the participant touches any object other than the moving object.
- ***object***: There will be one of these properties for each touchable object in the scene. To establish whether the object is interactive (can be moved) or not.
  - ***objectValue***: The value associated with the object. Decimal value.

keyboard

When the scene ends, a keyboard appears on the screen and the user can type the response.

- ***keyboardType***: Standard or numeric keyboard.
- ***responseInTitle***: When the results report is generated, we can make the value of the keyboard response appear in the section title. For example, if we request a participant name or code, it may be helpful to have that code in the title of the corresponding section of the result to easily identify the participant.

keys



If a hardware keyboard is available, it is possible to link some keys to certain response values so the user can respond simply by pressing a key at any time.

- ***startTime***: Time from which it is possible to respond. By default it is zero, the user can respond at any time. Time property.
- ***endTime***: Maximum time until which it is possible to respond. Time property.
- ***wrongTiming***: If 0 it is not possible to respond before startTime or after endTime. If 1 it is possible to respond before startTime or after endTime but the response is considered incorrect. Integer value 0 or 1.

END OF TRIAL CONDITIONS

End of trial conditions

In the ***section*** menu, you can create conditions that are evaluated after each trial.
A trial is the presentation of all the scenes of a section consecutively.
The order of the conditions in the menu is the order in which they are evaluated.
If a condition is true, the action associated with that condition will be performed and the following conditions will not be evaluated.

Possible conditions:

- when the number of trials = n
- when the number of trials responded in time = n
- when the number of trials not responded in time = n
- when the number of correct trials = n
- when the number of incorrect trials = n
- when the last trial was responded in time
- when the last trial was not responded in time
- when the last trial was correct
- when the last trial was incorrect
- when all the section trials have been performed

Possible actions:

- move to another section
- end the test

If neither condition is true, a new trial in the same section is performed.

Practical case

Suppose you have a test consisting of 100 trials.
You want to present a message when the participant has performed half of the trials.
You want to present another message to encourage the participant when the number of incorrect responses reaches a certain value.

In this particular example you need 3 sections:

- ***sectionMiddle***: containing a scene with a text message: “You are halfway through the test!”. It consists of a single trial.
- ***sectionWrong***: containing a scene with a text message: “Please concentrate. You can do better!”. It consists of a single trial.
- ***sectionMain***: contains your main scenes and consists of 100 trials.

In sectionMain you need the following conditions:

- when the number of trials = 50: ***sectionMiddle***
- when the number of incorrect trials = 30: ***sectionWrong***
- when the number of incorrect trials = 50: ***sectionWrong***
- when all the section trials have been performed: ***End the test***

In sectionMiddle and sectionWrong you need a single condition:

- when all the section trials have been performed: ***sectionMain***

SELECTION OF VARIABLES

Variable and Values

The name of the variable and the list from which it takes the values.

Related Variables

You can add and remove related variables according to whether you want them to have the same selection method as the first variable or not.
The number of possible values for all the related variables must be the same. Therefore, all the lists from which these variables obtain values must have the same number of elements.

Selection Method

The different methods used for the variables to get their values from their list.

- ***all values in order***: The variable gets all the values from the list in order.
  - ***selectionMethodPriority***: The priority of the method.
    For example:
    First variable gets its values in order and its possible values are: 0, 1.
    Second variable also gets its values in order and its possible values are 10, 20, 30.
    Depending on which variable has a higher priority the result can be:
    - 0,10
    - 0,20
    - 0,30
    - 1,10
    - 1,20
    - 1,30

      if the first variable has higher priority, or:
    - 0,10
    - 1,10
    - 0,20
    - 1,20
    - 0,30
    - 1,30

      if the second variable has higher priority.
  - ***alternatedVariable***: There can only be one variable (or group of variables) of alternate type. This variable goes through all its values (in order or randomly) before starting with another repetition of all its values.

- ***all values in random order***: The variable gets all the values from the list in random order.
  - ***alternatedVariable***: There can only be one variable (or group of variables) of alternate type. This variable goes through all its values (in order or randomly) before starting with another repetition of all its values.

- ***one fixed value***: The variable value is always the same fixed value from the list.
    - ***positionValue***: Select one of the values from the list. From 1 to the total number of values.

- ***one random value***: In each trial, the variable value is a random value from the list.
    - ***selectionMethodEqual***: Whether or not the selection of values for all the related variables is the same.
        - ***equal***: All the related variables take a value that is in the same position in their respective lists.
        - ***different***: All the related variables take a value that is in a different position in the list for each variable.
          The number of variables must be less or equal than the number of possible values in the lists to meet this condition.

- ***depends on correct***: The variable value depends on the correction of the previous response.
    - ***correct/incorrect:*** After a correct response, the variable value is the first value from its list.
      After an incorrect response, the variable value is the second value from its list.
      If there has been no response yet, the previous response is considered correct. So in case the variable is used before giving any response, the variable value is the first value from its list.
    - ***1up/1down***: After a correct response, the variable steps down, to the previous value in its list.
      After an incorrect response, the variable steps up, to the next value in its list.
    - ***1up/2down***: After two consecutive correct responses, the variable steps down, to the previous value in its list.
      After an incorrect response, the variable steps up, to the next value in its list.
      To get to the relevant values faster, a 1up/1down method is applied until the first incorrect response, then the method changes to 1up/2down.
    - ***1up/3down***: After three consecutive correct responses the variable steps down, to the previous value in its list.
      After an incorrect response, the variable steps up, to the next value in its list.
      To get to the relevant values faster, a 1up/1down method is applied until the first incorrect response, then the method changes to 1up/3down.
    - ***initialValue***: Select one of the values from the list. From 1 to the total number of values. This will be the first value. From there, the value will change following the chosen formula.

Blocks

When the values of a variable are in a list of Blocks there is no selection method. The list of blocks manages how the values are chosen.

Examples of use

Using one block

Suppose you want the values of a particular variable to alternate between two lists in each trial. For example you want that if the value in the n trial is positive, the value in the n+1 trial is negative and vice versa.
First create the two lists of numerical values:

- In the first list the values are: 1, 2, 3, 4.
- In the second list the values are: -1, -2, -3, -4.

To make the alternation possible you need to create a list of blocks. And set the following values:

- Repetitions
  - numberOfBlocks = 1
  - lengthOfBlocks = 10
- Types of blocks
  - typesOfBlocks = 1
- Block
  - firstList = your first list of values
  - secondList = your second list of values
  - startingList = random
  - probChangeList = 1

When a variable takes its values from a list of blocks, it is assigned an initial list, in this case, one of the two lists is chosen randomly. After each trial, the list changes with a certain probability (in this example with probability one). For each trial, the value of the variable is chosen randomly from the values in the corresponding list.
If you preview the variable values, you will get something similar to:
-2, 4, -4, 1, -1, 2, -3, 1, -3, 4

Using two blocks

Suppose you want the values of a particular variable to be sometimes in repetition blocks and other times in alternation blocks. This means that in certain blocks it is more likely to repeat the same list and in others it is more likely to alternate it.

First create two lists of numerical values:

- In the first list the values are: 1, 2, 3, 4.
- In the second list the values are: -1, -2, -3, -4.

Then, create the list of blocks. And set the following values:

- Repetitions
  - numberOfBlocks = 5
  - lengthOfBlocks = 10
- Types of blocks
  - typesOfBlocks = 1
  - startingBlock = random
  - probChangeBlock = 1
- First Block
  - firstBlockFirstList = your first list of values
  - firstBlockSecondList = your second list of values
  - firstBlockStartingList = random
  - firstBlockProbChangeList = 0.2
- Second Block
  - secondBlockFirstList = your first list of values
  - secondBlockSecondList = your second list of values
  - secondBlockStartingList = random
  - secondBlockProbChangeList = 0.8

The first block is chosen randomly between the repetition block (with probability of changing list = 0.2) and the alternation block (with probability of changing list = 0.8).
Blocks consist of 10 trials and the probability of changing from one block to the other is 0.5, that means that the type of block is chosen randomly when the previous block ends.
If you preview the variable values, you will get something similar to:

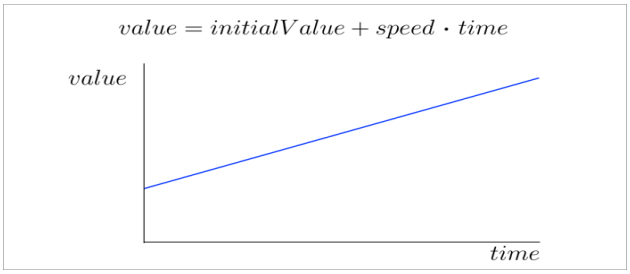
2, 2, 1, 4, -2, -2, -4, -1, -2, -3 repetition block
-1, -3, -2, 1, 4, 4, 4, 1, 4, 4 repetition block
-3, 4, -1, 4, -4, 2, -2, 1, -3, 3 alternation block
1, 2, -2, -3, -3, -4, -2, -3, -2, -2 repetition block
4, -2, 3, -2, 1, 4 ,-3, 3, -3, -4 alternation block

TIME DEPENDENCY

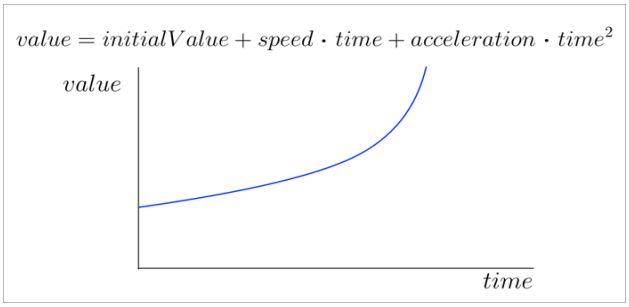
Most properties can change over time by fitting one of the following equations:

linear

The value changes linearly with time.

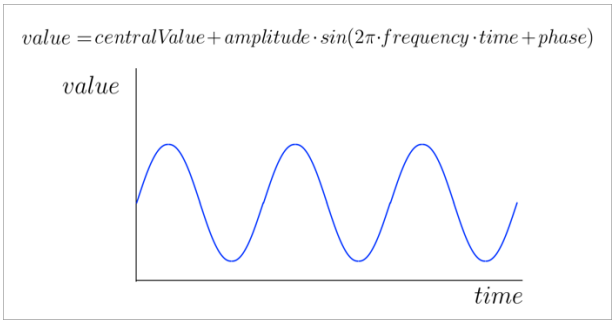


quadratic



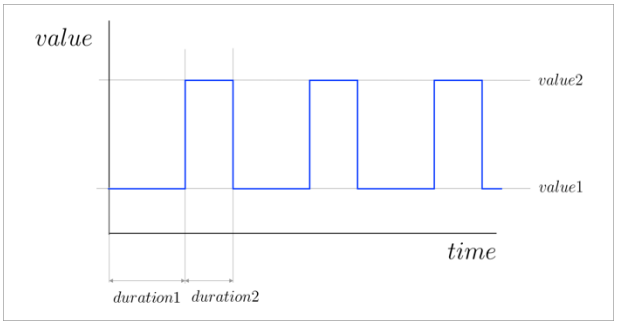
The value changes quadratically with time.

sinusoidal



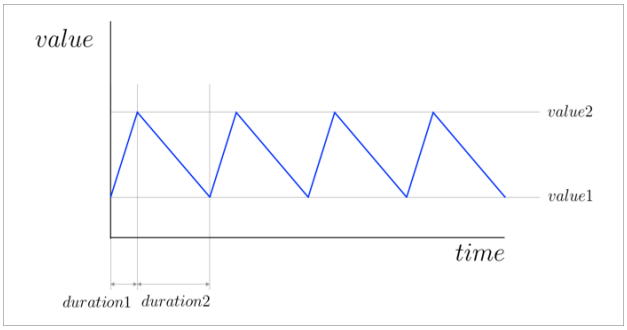
The value oscillates sinusoidally with time.

rectangle wave



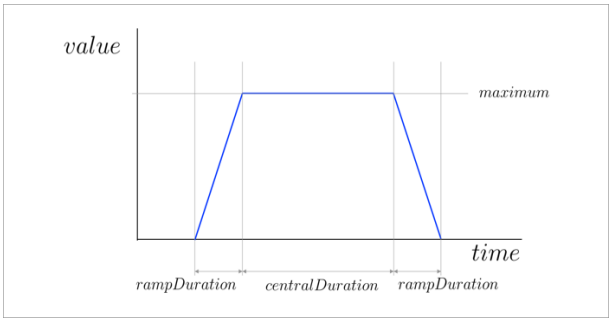
The value follows a cycle where it is equal to value1 during duration1 and equal to value2 during duration2.

triangle wave



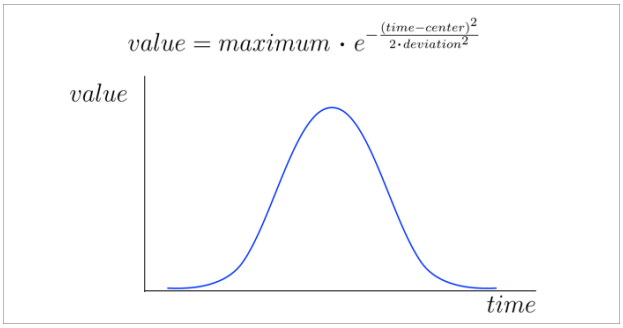
The value follows a cycle where it changes linearly from value1 to value2 during duration1 and changes from value2 to value1 during duration2.

pulse



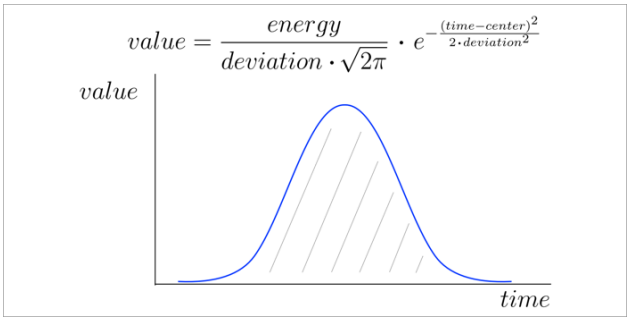
The value changes linearly from zero to maximum during rampDuration. It is equal to the maximum for centralDuration.
It changes linearly from maximum to zero during rampDuration.

gaussian



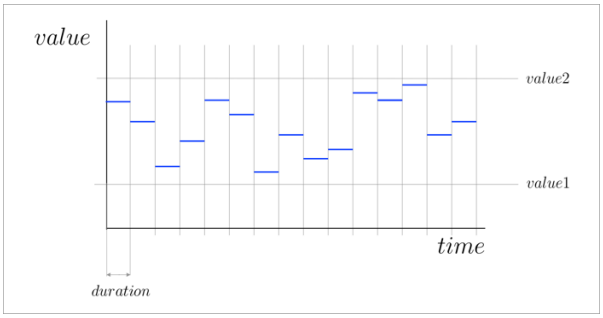
The value follows a gaussian function.

gaussian energy



The value follows a gaussian function normalised to have always the same energy (area under the curve) regardless of the standard deviation.

random



Random values chosen in the interval from value1 to value2.
The duration is the same for each of them.

MEASUREMENT UNITS

Size

Size can be measured in pixels, centimetres, inches, visual degrees, screen width units or screen height units.
To transform inches or centimetres to pixels we use the ppi value from the app settings.
To transform visual degrees to pixels we use the viewingDistance value from the test properties.
Screen width units are measured in a scale -1 to 1. We use the screenResolution value from the app settings (width in landscape mode).
Screen height units are measured in a scale -1 to 1. We use the screenResolution value from the app settings (height in landscape mode).

Time

Time can be measured in seconds or frames.
To transform frames to seconds we use the frameRate value from the test properties.

Angle

Angles can be measured in radians or degrees.

Luminance

Luminance is measured in a scale from 0 to 1. Using the value of maximumLuminance of your device and the value of the luminance in your test settings, the real value in cd/m2 is also displayed informatively.

TIMING AND FRAMES

Timing and Frames

The screen frame rate can be 60 or 120 Hz depending on the device and the test properties. That means that a new image has to appear on the screen every 16.667 or 8.889 milliseconds. That time is the expected frame duration.

The calculations required to render a new image may take longer than the expected frame duration. In that case, the previous image will be on the screen for a longer time, until all the calculations are made and a new image can appear. We call the frames that last more than expected long frames.

Ideally, the number of long frames in your test should be zero. But because the number of calculations is higher during some frames (for example during the frame that appears just after a response) it is possible that the total number of long frames in your test is not exactly zero, although it should be a very small percentage of the total.

Every time you preview a test, a message is displayed informing you about the number of long frames.

Every time you run a test, a results report is created. There you get detailed information about which frames last longer than expected and their real duration in seconds.

If you get more than a few long frames, you need to reduce the computational cost of your test to get a more steady frame rate. Try to follow the next steps:

- Reduce the frame rate to 60 Hz if you are running your test at 120 Hz and having this high frame rate is not essential to your test.
- Minimize the numberOfLayers in your scenes because each extra layer adds additional computational cost.
- Disable the continuosResolution property of the scene.
- Reduce the number of stimuli present at the same time.
- Reduce the size of the stimuli.
- If you want an object to be present in some trials but not in others, make its activated property zero or one depending on the trial. When the activated property is zero, neither CPU or GPU cycles are used to compute the object.

Image and audio synchronization

When the user specifies that the auditory and visual signals should be presented at the same, some small audiovisual delays occur. The average delay is around -10 to 10 ms depending on the device. It is possible to correct this average delay.
You need to present several times an audiovisual signal specified to be presented at the same time and calculate the average delay measuring the signals with an oscilloscope. Then, you should include the average delay in the ***delayAudio60*** or ***delayAudio120*** variable of the app settings using a positive sign if you want that the correction delays the auditory signal presentation and a negative sign if you want that the correction delays the visual signal presentation.
The variability of the delay across presentations (precision) is less than 1 millisecond (standard deviation) and cannot be corrected.

Reaction times

The images to be presented on the screen are delivered with a certain constant frame rate.
From the moment the images are ***delivered*** until they are ***displayed*** on the screen, there is a small ***delay*** of about 10-30 ms depending on the hardware. This delay is approximately constant for each session although variable across sessions.

All timers are synchronized with the ***delivery time*** of the images and the delay in the presentation of the images is stored in a variable named ***scene_delayDisplay*** in the results report.

When measuring reaction times, you may want to consider this delay. If you want to know the real duration that the image has been drawn on the screen at the time of the response, it is necessary to subtract the value of ***scene_delayDisplay*** from the value of ***scene_responseTime***.

If you want to know the real duration of a scene, you can use the variable scene_duration, no need for any correction, as all the frames that are part of the scene are displayed with the same delay.

These are the relevant values you get in the results report:

- ***scene_duration***: the real duration of the scene in seconds (one value for each scene in the section).

- ***scene_responseTime***: the real time of the response, if there is one. Note that the duration of the scenes that require a response is usually one frame (16.666 or 8.888ms) longer than the scene_responseTime. This additional frame is used to stop any sound or video and prepare for rendering the next scene. When the scene does not require a response, its duration is known in advance, and these processes and calculations are performed in the last frame, so in that case, no additional frame is needed.

- ***scene_delayDisplay***: the images to be presented on the screen are delivered with a certain constant frame rate. From the moment the images are ***delivered*** until they are ***displayed*** on the screen, there is a small ***delay*** of about 10-30 ms depending on the hardware. This delay is approximately constant for each session although variable across sessions. This variable measures that delay. Thus, if you want to know the real duration that the image has been drawn on the screen at the time of the response, it is necessary to subtract the value of ***scene_delayDisplay*** from the value of ***scene_responseTime***.

When working with reaction times, you will need also to consider the sampling rate of the touch responses.
Touch information is sampled at 120 Hz in all devices, except for the iPad Pro 11-inch first generation (and later) and the iPad Pro 12.9-inch third generation (and later) in which the sampling rate is 240 Hz.

RESULTS REPORT

Results report

In the Results menu, you can visualize or email the results of the performed tests. In addition to the general report, which is saved as a .txt file, a .csv file is generated for each section, containing the information for all the variables and responses in that particular section.

You can visit the Import and Export section to learn how to email your reports.

The reports are divided into the following parts:

Summary

Information about App Settings and Test Properties.

- test
- user
- date
- device
- system
- audioRate
- screenResolution
- ppi
- rampTime
- maximumLuminance
- frameRate
- luminance
- viewingDistance
- gamma
- randomness

Sections

For each section of the test, a list with all the variables of any object in any scene of the section is displayed.
Each line in the list is a different trial and the values in each line are comma-separated.
In addition to the variables, the following properties are also displayed (where appropriate):

- ***trial***: the trial number, starting at 1.

- ***scene_duration***: the real duration of the scene in seconds (one value for each scene in the section).

- ***scene_responseTime***: the real time of the response, if there is one. Note that the duration of the scenes that require a response is usually one frame (16.666 or 8.888ms) longer than the scene_responseTime. This additional frame is used to stop any sound or video and prepare for rendering the next scene. When the scene does not require a response, its duration is known in advance, and these processes and calculations are performed in the last frame, so in that case, no additional frame is needed.

- ***scene_delayDisplay***: the images to be presented on the screen are delivered with a certain constant frame rate. From the moment the images are ***delivered*** until they are ***displayed*** on the screen, there is a small ***delay*** of about 10-30 ms depending on the hardware. This delay is approximately constant for each session although variable across sessions. This variable measures that delay. Thus, if you want to know the real duration that the image has been drawn on the screen at the time of the response, it is necessary to subtract the value of ***scene_delayDisplay*** from the value of ***scene_responseTime***.

- ***scene_response***: the value of the response, if there is one.

- ***scene_position***: if the response is touch screen. It can be measured in ***scene_positionX*** and ***scene_positionY*** or ***scene_positionRadius*** and ***scene_positionAngle***.

- ***scene_finalPosition***: if the response is path or move object. It is the final position of the path. It can be measured in ***scene_finalPositionX*** and ***scene_finalPositionY*** or ***scene_finalPositionRadius*** and ***scene_finalPositionAngle***.

- ***trialValue***: the value associated with each trial in the section.

- ***correct***: 0 for incorrect trials, 1 for correct trials.

- ***respondedInTime***: if any of the scenes ends before an answer is given or the answer is given outside its allowed window time, the value of this variable is 0. Otherwise, it is 1.

When the response is ***path*** or ***move object***, a new list is created with the trajectories. Each line in the list is a different trial, the values in each line are comma-separated.

- ***trial***: the trial number.

- ***scene_positions***: it can be measured in ***scene_positionsX*** and ***scene_positionsY*** or ***scene_positionsRadius*** and ***scene_positionsAngle***. Positions are semicolon-separated.

- ***scene_times***: the time of each position in seconds. Times are semicolon-separated.

Values of the constant properties

The values of all the constant properties of all the objects in the test.

Frame rate

- Frame rate
- Frame duration
- Number of long frames
- Long frames and duration

Visit timing and frames for more information about timing.

IMPORT AND EXPORT

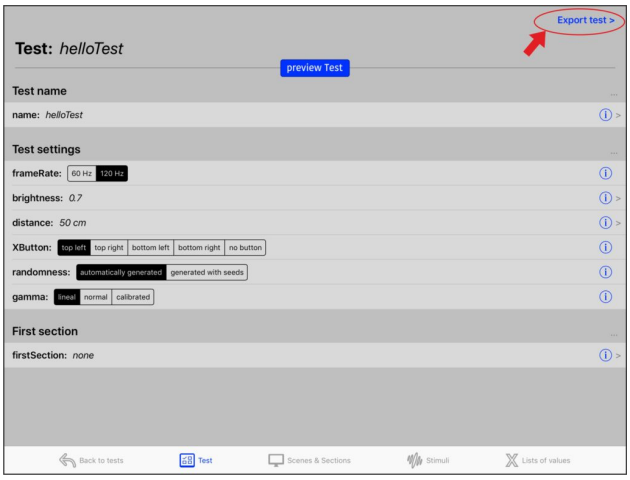
Export results

The reports generated in the results menu can be sent by email, simply by clicking the email button at the top of the screen. In addition to the general report, which is saved as a ***.txt*** file, a ***.csv*** file is generated for each section, containing the information for all the variables and responses in that particular section.

Export test

Once a test is created, it can be exported to any other device with StimuliApp installed (for example, a test created on an iPad could be run on an iPhone). To export a test, ***Export test*** should be pressed in the ***Test*** menu. The user will be able to email a ***.stimulitest*** file containing a ***.json*** description of the parameters of the test to any device. To import the test in the receiver device, the ***.stimulitest*** file should be opened with StimuliApp (if StimuliApp does not appear in the list of applications the user should click the ***More*** option).

The ***.***.stimulitest*** files, as they contain a ***.***.json*** description, could be edited with an IDE. By default, the files are generated in a single line, but many editors have an automatic option to change it to multiple lines. We think, however, that it is easy to get lost in the structure of a *.json* file and recommend the modification of tests within StimuliApp.

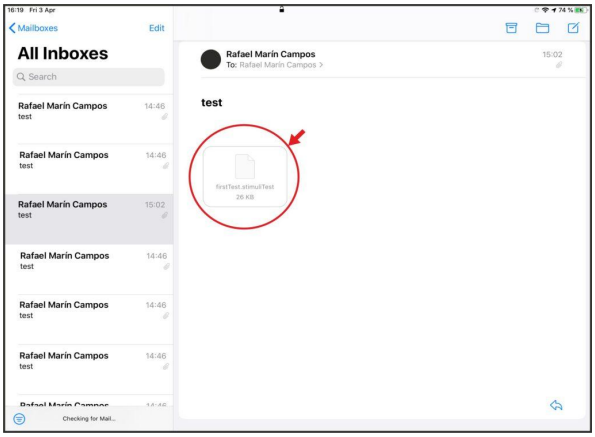


Import test

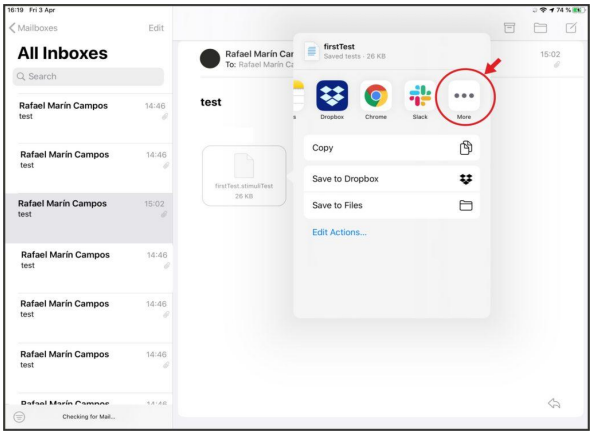
If you receive a .***stimulitest*** file, you can just select the file and open it with ***StimuliApp***.
StimuliApp will launch and the new imported test will appear in the ***Tests*** menu.
If you import a test created on a different device, any list of images, audios or videos will have to be recreated with the actual files stored in the library of the new device.

Open a file sent by email

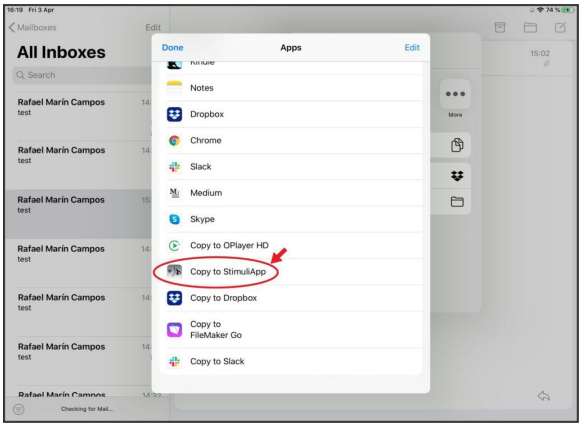
If you receive the test by email, open the ***Mail*** app on your device and click on the file.



A new menu appears where you can select StimuliApp from the list of suggested applications. If StimuliApp does not appear in the list, click the ***More*** option.



More applications appear in the new menu. Select ***Copy to StimuliApp***.

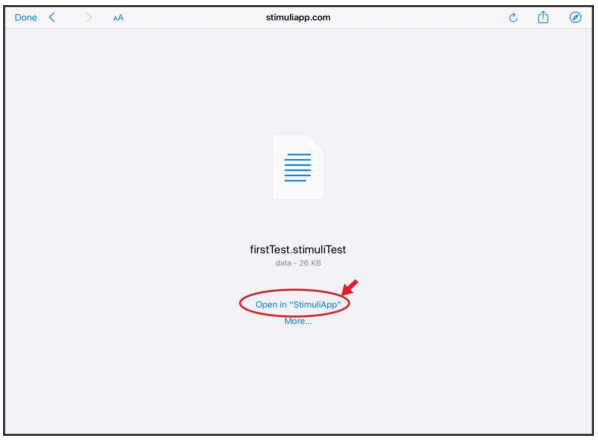


StimuliApp will launch and the new imported test will appear in the ***Tests*** menu.

Open a file stored in the Files application

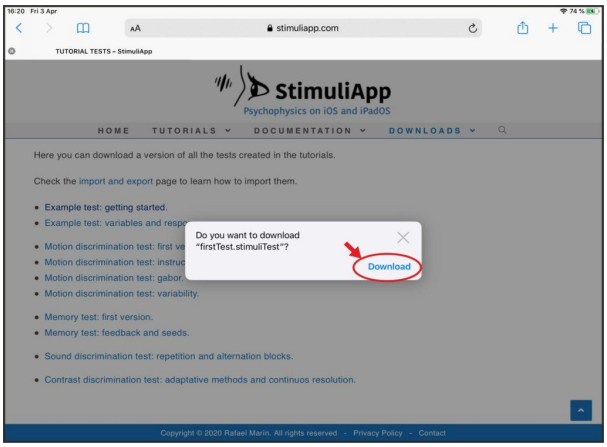
You can open any test stored in your ***Files*** application.

Open the Files app and click on the file, a new menu will appear, just click ***Open in StimuliApp***, then close the window and start ***StimuliApp*** to find the new test in the ***Tests*** menu.



Download a file from Safari

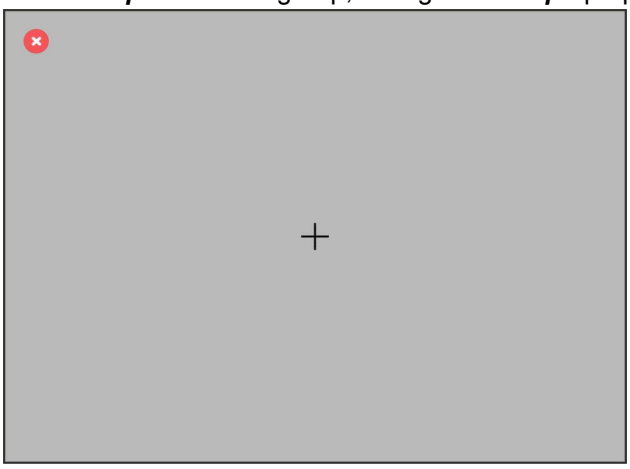
You can get tutorial tests from the download menu on this page.
Click on the test and the option to copy to ***StimuliApp*** should appear.
It may happen that Safari only gives you the option to download the file. Don't worry, just download it and then go to your ***Files*** app to copy the file to ***StimuliApp***.



HOW TO: CREATION OF COMMONLY USED STIMULI

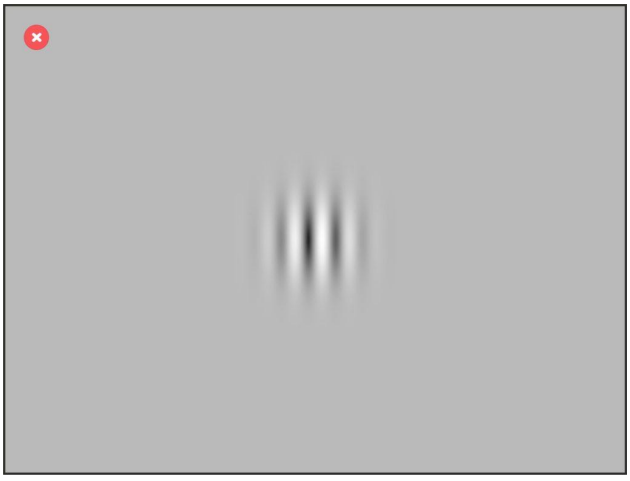
Fixation cross

Create a new stimulus.
In the ***Shape and size*** group, change the ***shape*** property to ***cross***.



Gabor

Create a new stimulus.
In the ***Stimulus type*** group, change the ***type*** property to ***grating***.
In the ***Shape and size*** group, change the ***size*** property to 600 pixels.
In the ***Contrast*** group, change the ***contrast*** property to ***gaussian***.

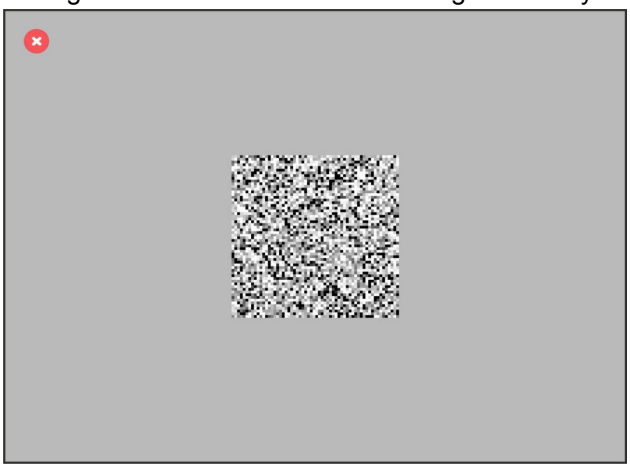


Moving gabor

Starting with the previous gabor stimulus.
In the ***Stimulus type*** group, change the ***phase*** property to ***time dependent*** with a ***linear*** function.
Change the ***phaseSpeed*** property to 30 rad/s.

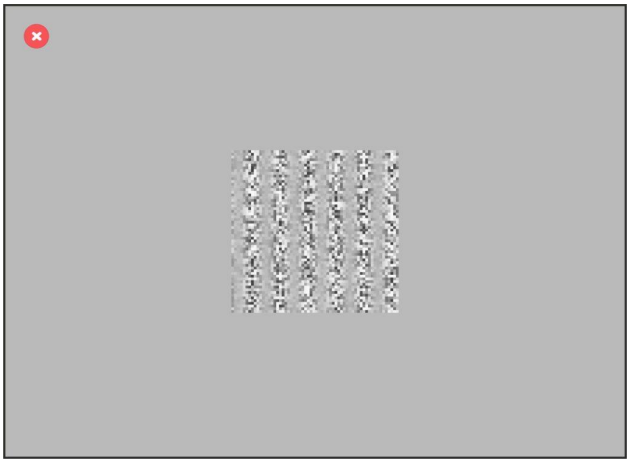
White noise image

Create a new stimulus.
In the ***Stimulus type*** group, change the ***color*** property to ***luminance***.
Change the ***colorLuminance*** property to 0.5.
In the ***Shape and size*** group, change the size property to 600 pixels.
In the ***Noise*** group, change the noise property to ***gaussian***.
Change the ***noiseTimePeriod*** to a large value if you want static noise.



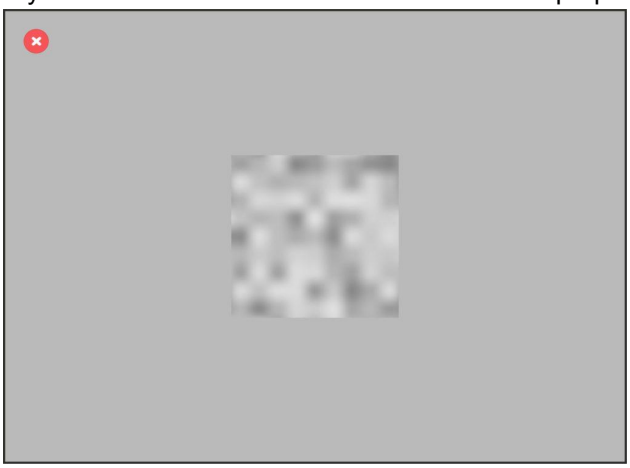
Contrast-Modulated Sine Wave on a Noise Texture Carrier

Starting with the previous white noise stimulus.
In the ***Modulator of contrast*** group, change the ***modulator*** property to ***sinusoidal***.
Change the ***modulatorAmplitude*** to 1.
Change the ***modulatorPhase*** property to ***time dependent*** with a ***linear*** function.
Change the ***modulatorPhaseSpeed*** property to 10 rad/s.



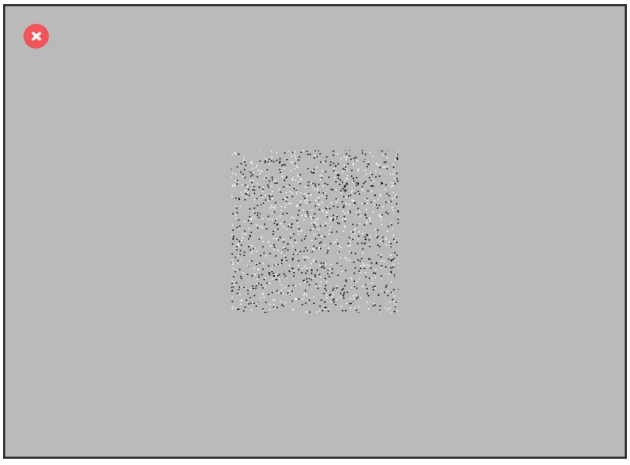
Perlin noise

Create a new stimulus.
In the ***Stimulus type*** group, change the ***color*** property to ***luminance***.
Change the ***colorLuminance*** property to 0.5.
In the ***Shape and size*** group, change the ***size*** property to 600 pixels.
In the ***Noise*** group, change the ***noise*** property to ***perlin***.
Try different values for the ***noiseSmoothness*** property to see the differences.



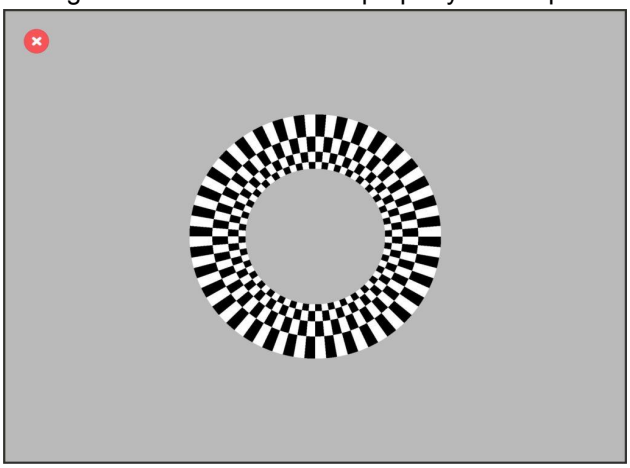
Random Dots

Create a new stimulus.
In the ***Stimulus type*** group, change the ***type*** property to dots.
In the ***Shape and size*** group, change the ***size*** property to 600 pixels.
Change any of the properties to change the ***diameter***, ***direction*** and ***color*** for the ***type1*** and ***type2*** dots.



Ring of checkerboards

Create a new stimulus.
In the ***Stimulus type*** group, change the ***type*** property to ***radialCheckerboard***.
Change the ***boxAngleSize*** to 5 degrees.
Change the ***diameter1*** property to 550 pixels.
Change the ***diameter2*** property to 625 pixels.
Change the ***diameter3*** property to 737 pixels.
In the ***Shape and size*** group, change the ***shape*** to ring.
Change the ***exteriorDiameter*** property to 900 pixels.
Change the ***interiorDiameter*** property to 500 pixels.



PROPERTY TYPES

color

Measured with a decimal value from 0 to 1.
It can only be dependent on time when it is expressed as: RGB vars or luminance.
can be expressed as:

- ***vector3d***: One only variable with 3 components (r;g;b).
- ***RGB vars***: 3 independent variables, one for each color: red, green and blue.
- ***luminance***: One only variable (r=g=b) for grayscale colors. The real value in cd/m2 is also provided.

size

Measured in pixels, centimetres, inches or visual degrees.

2dsize

Measured in pixels, centimetres, inches or visual degrees.
It can only be dependent on time when it is expressed as: cartesian vars or x=y.
can be expressed as:

- ***vector2d***: One only variable with 2 components (horizontal;vertical).
- ***cartesian vars***: 2 independent cartesian variables.
- ***x=y***: One only variable to measure size making sizeX = sizeY.

position

Measured in pixels, centimetres, inches or visual degrees.

2dposition

Measured in pixels, centimetres, inches or visual degrees. X increases from left to right and Y increases from bottom to top.
It can only be dependent on time when it is expressed as: cartesian vars or polar vars.
can be expressed as:

- ***vector2d***: One only variable with 2 components (horizontal;vertical).
- ***cartesian vars***: 2 independent cartesian variables.
- ***polar vars***: 2 independent polar variables.

angle

Measured in radians or degrees. Measured counterclockwise from the x axis.

time

Measured in seconds or frames. The duration of a frame depends on the frame rate property of the test.
When the frame rate is 60Hz, the duration of a frame is 16,666 milliseconds.
When the frame rate is 120 Hz, the duration of a frame is 8,333 milliseconds.

frequency

Frequency measured in hertz (cycles per second).

font

Any of the system fonts.

positive non-zero integer

Positive non-zero integer.

integer value 0 or 1

The only possible values are 0 or 1.

integer from 3 to 10

Integer value from 3 to 10. Used to create polygons with a fixed number of sides.

value from 0 to 1

Decimal value from zero to one.

positive (or zero) decimal value

Positive (or zero) decimal value.

decimal value

Decimal value.