Psychopy: a Python library for psychological experiments. Session 2 – PsychoPy Introduction

Jaime A. Riascos Salas

Computational Neurosciences and Cybernetics Systems (CONCYS)

Institute of Informatics

Federal University Rio Grande do Sul (UFRGS)

Porto Alegre – Brazil.

.inf INSTITUTO DE INFORMÁTICA UFRGS

1. Introduction PsychoPy
2. Installation
3. Architecture – Builder and Coder
4. Understanding PsychoPy
5. Monitor object
6. Stimuli objects
7. Hands-on!!!

INSTITUTO DE INFORMÁTICA UFRGS

PsychoPy
Psychology software in Python

PsychoPy

What is it ?

It is an open-source package for creating and running psychological experiments. PsychoPy uses both OpenGL (most widely adopted 2D and 3D graphics library) and Python to give scientists and students a free and simple stimulus presentation and control package. It is widely used by many labs worldwide for psychophysics, cognitive neuroscience and experimental psychology.

www.psychopy.org

.inf INSTITUTO DE INFORMÁTICA UFRGS

PsychoPy
Psychology software in Python

PsychoPy

Advantages?

Instead the paid software (Matlab, up 600 $USD; e-Prime, up 300 $USD), PsychoPy is totally free and open-source.

As any open source software, you are able to download and modify the package as much as you want. PsychoPy is developed in a single piece of software enabling to be precise enough for psychophysics, intuitive enough for undergraduate psychology and flexible enough for anyone who is intending to develop own experiments.

.inf INSTITUTO DE INFORMÁTICA UFRGS

PsychoPy
Psychology software in Python

PsychoPy

Features?

- Simple install process;
- Precise timing;
- Huge variety of stimuli (see screenshots) generated in real-time:
  - Images
  - Random dots
  - Text (unicode in any truetype font)
  - Shapes and movies (DivX, mov, mpg…)
  - sounds (tones, numpy arrays, wav, ogg…)

INSTITUTO DE INFORMÁTICA UFRGS

PsychoPy
Psychology software in Python

PsychoPy

Features?

- Platform independent - run the same script on Win, macOS or Linux;
- Flexible stimulus units (degrees, cm, or pixels);
- Input from keyboard, mouse, microphone or button boxes;
- Multi-monitor support;
- Automated monitor calibration (for supported photometers);

- **Coder interface**, for those that like to program
- **Builder interface**, for those that don't

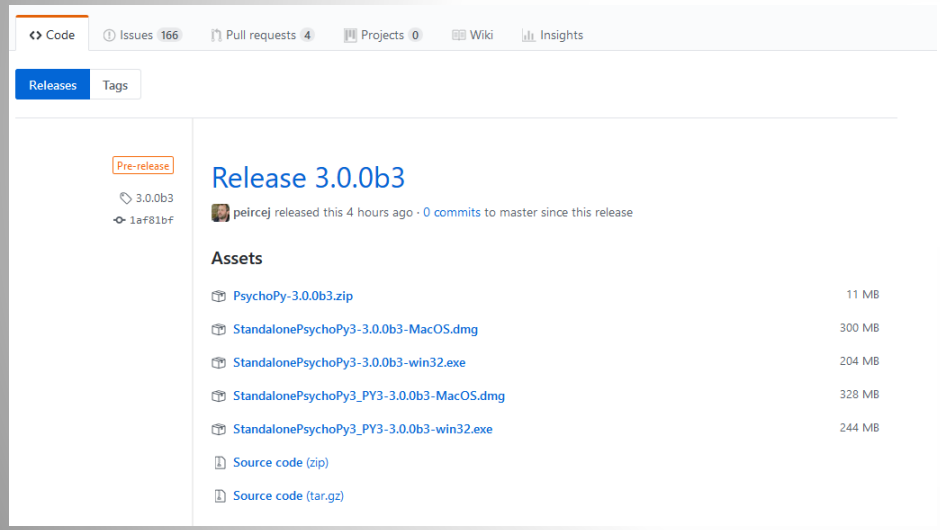
.inf INSTITUTO DE INFORMÁTICA UFRGS

PsychoPy
Psychology software in Python

PsychoPy - Installation

Windows/Mac users

Install the Standalone package:



Linux users

- sudo apt-get install psychopy (maybe you need to install some dependencies)
- Neurodebian (http://neuro.debian.net/)

http://psychopy.org/installation.html

INSTITUTO DE INFORMÁTICA UFRGS

PsychoPy
Psychology software in Python

PsychoPy - Installation

Manual Installation (win, linux, mac)

Now that most python libraries can be install using *pip* it's relatively easy to manually install PsychoPy and all it's dependencies to your own installation of Python. That isn't the officially-supported method (because we can't track which versions of packages you have) but for many people it's the preferred option if they use Python for other things as well.

Dependencies

You need a copy of Python 2.7.x from here, wxPython and probably pyo (or use an alternative audio library listed below). None of these support *pip install* yet so you need to download them:

- Python itself: http://www.python.org/download/ (**version 3.x is not supported yet**)
- wxPython: https://wxpython.org/download.php
- pyo audio: http://ajaxsoundstudio.com/software/pyo/
- PyQt4 or PyQt5 are handy but not required and need manual installation

Then, if you want **everything** available you could paste this in to your terminal/commandline and go and get a coffee (will take maybe 20mins to download and install everything?):

```
pip install numpy scipy matplotlib pandas pyopengl pyglet pillow moviepy lxml openpyxl xlrd configc
pip install pyserial pyparallel egi iolabs
pip install pytest coverage sphinx
```

Needed on Windows:

```
pip install pypiwin32
```

Needed on macOS:

```
pip install pyobjc-core pyobjc-framework-Quartz
```

http://psychopy.org/installation.html

INSTITUTO DE INFORMÁTICA UFRGS

PsychoPy
Psychology software in Python

PsychoPy Builder — For non-programmer users

PsychoPy Coder — For programmer users

PsychoPy Library — Packing like a Matlab Toolbox

- Stimuli
- Timing
- Trial

Python — Programming interface

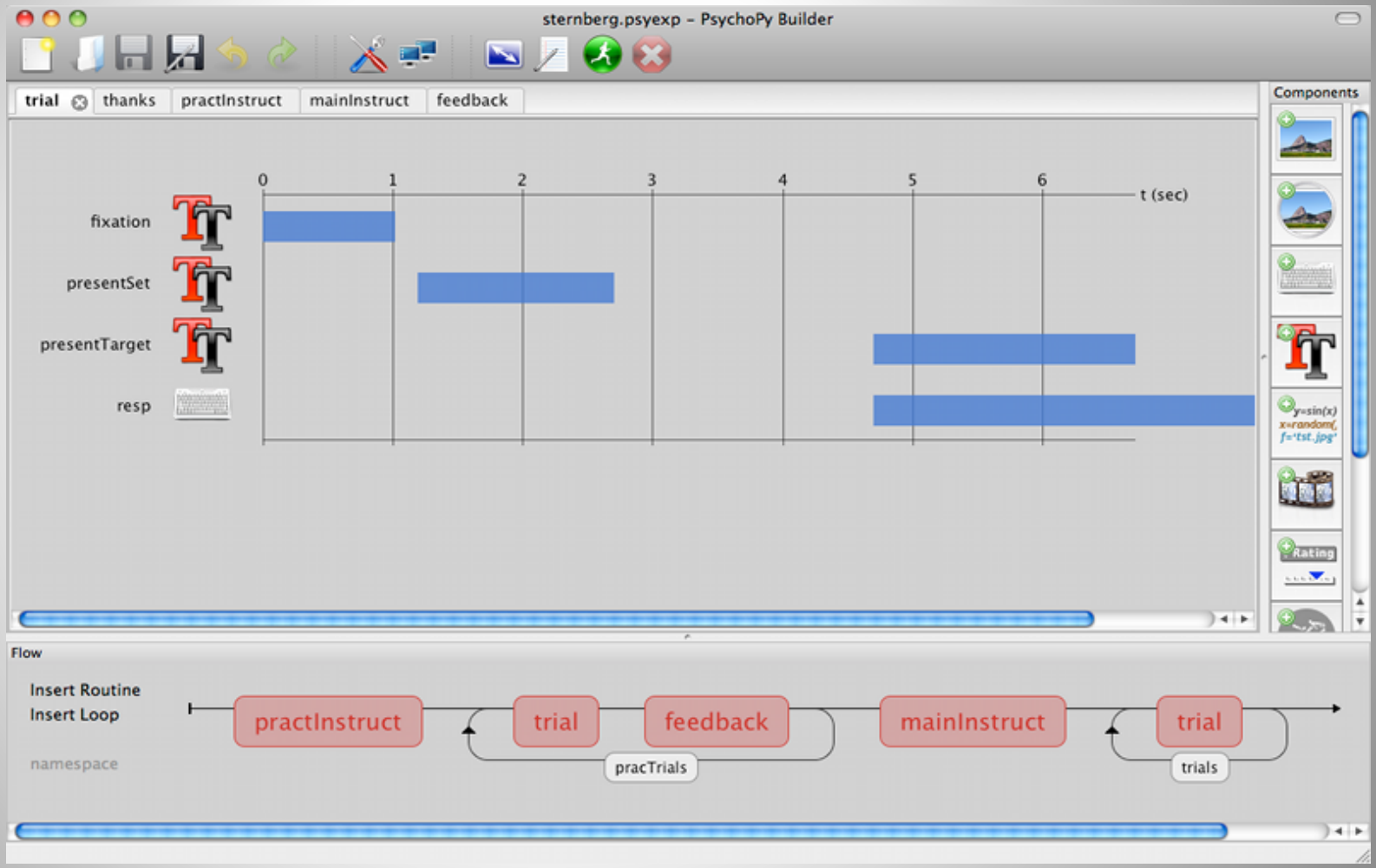
Low-level libraries

- OpenGL
- Maths (Numpy)
- User Interface (wxPython)

.inf INSTITUTO DE INFORMÁTICA UFRGS

PsychoPy
Psychology software in Python

PsychoPy Builder



PsychoPy
Psychology software in Python

PsychoPy Builder

Components

A right-placed panel where you can find all objects for creating stimulus and adding an input/output device. A default list of components content:

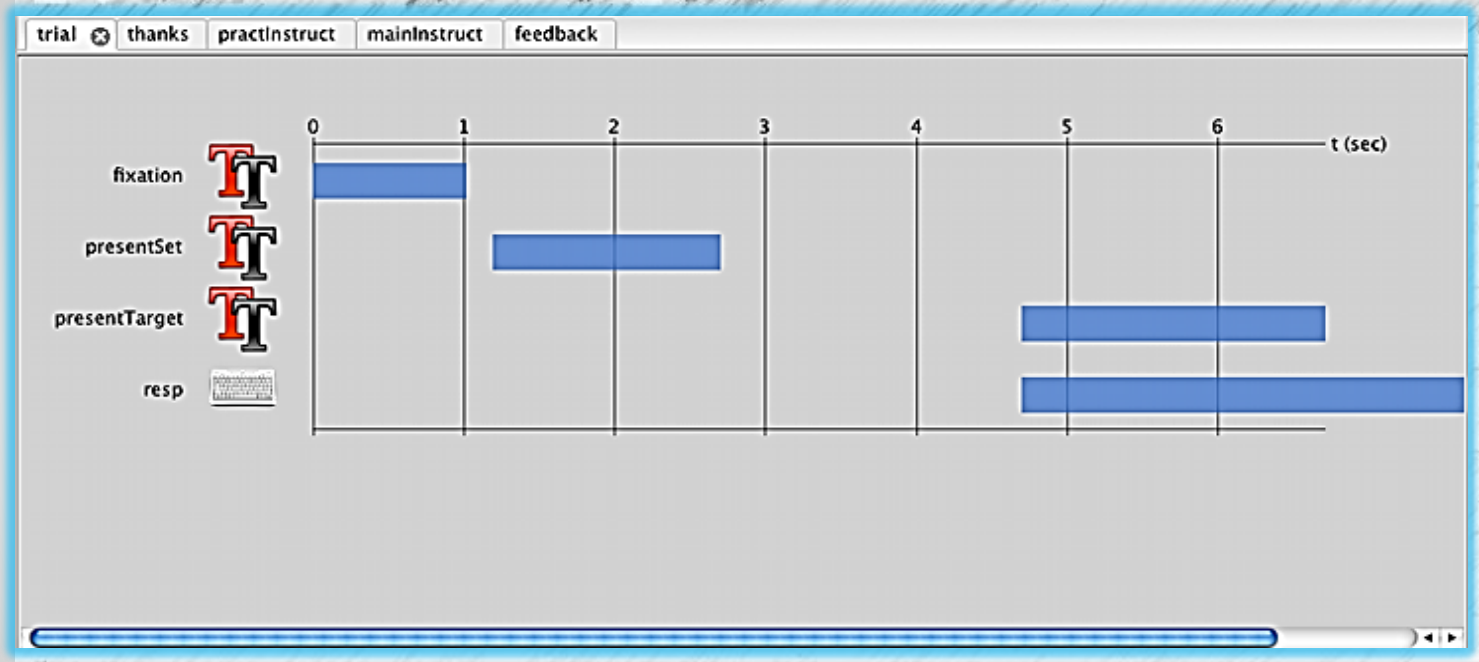
- Aperture Component
- Cedrus Button Box Component
- Code Component
- Dots (RDK) Component
- Grating Component
- Image Component
- ioLab Systems buttonbox Component
- Keyboard Component
- Text Component
- Variable Component
- Microphone Component
- Mouse Component
- Movie Component
- Parallel Port Out Component
- Patch (image) Component
- Polygon (shape) Component
- RatingScale Component
- Sound Component
- Static Component

INSTITUTO DE INFORMÁTICA UFRGS

PsychoPy
Psychology software in Python

PsychoPy Builder

Routines



A middle-placed panel where you can find any number of Routines used in the experiment, describing the timing of stimuli, instructions and responses. These are portrayed in a simple track-based view, similar to that of video-editing software, which allows stimuli to come on go off repeatedly and to overlap with each other.

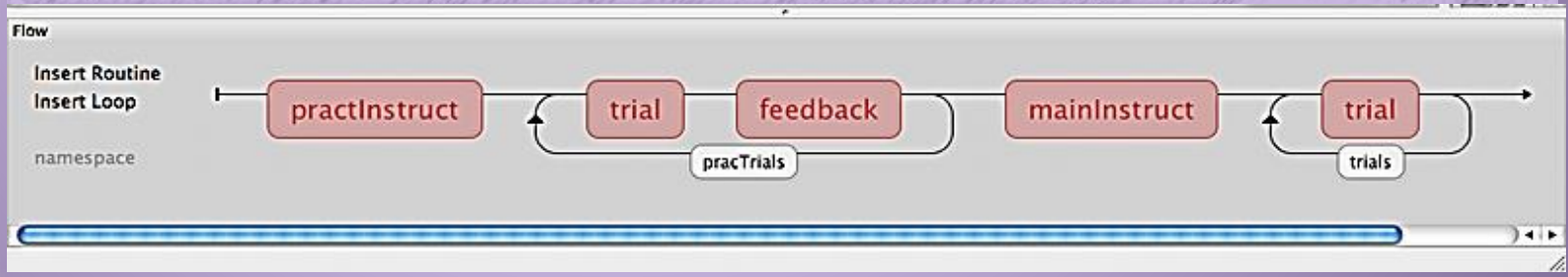
.inf INSTITUTO DE INFORMÁTICA UFRGS

PsychoPy
Psychology software in Python

PsychoPy Builder

Flow

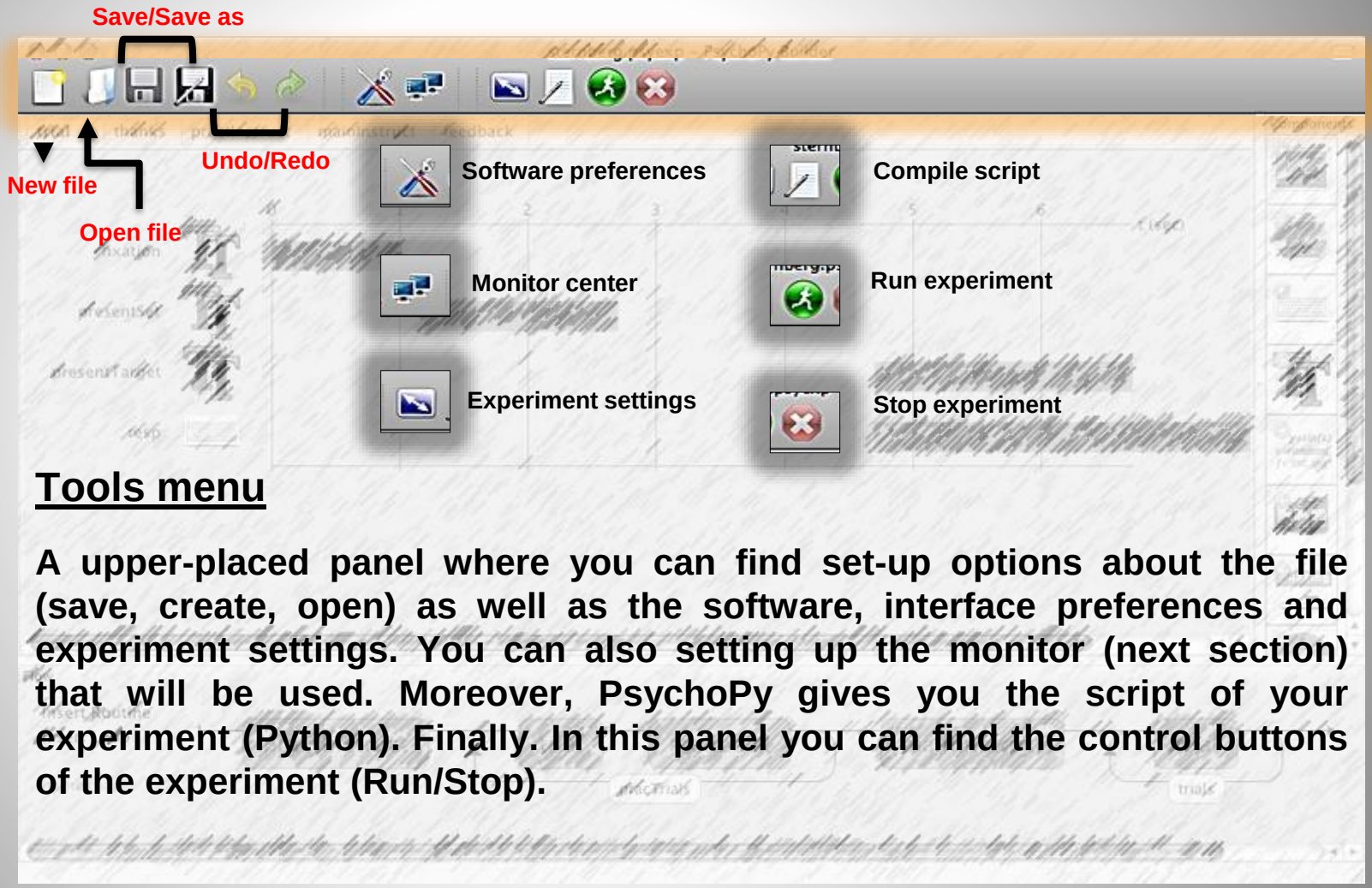
A bottom-placed panel where you can control all routines and loops used to form an experiment. All experiments have exactly one Flow. This takes the form of a standard flowchart allowing a sequence of routines to occur one after another, and for loops to be inserted around one or more of the Routines. The loop also controls variables that change between repetitions, such as stimulus attributes.



INSTITUTO DE INFORMÁTICA UFRGS

PsychoPy
Psychology software in Python

PsychoPy Builder



Tools menu

A upper-placed panel where you can find set-up options about the file (save, create, open) as well as the software, interface preferences and experiment settings. You can also setting up the monitor (next section) that will be used. Moreover, PsychoPy gives you the script of your experiment (Python). Finally. In this panel you can find the control buttons of the experiment (Run/Stop).

.inf INSTITUTO DE INFORMÁTICA UFRGS

PsychoPy
Psychology software in Python

What are the main steps to create any experiment using PsychoPy?

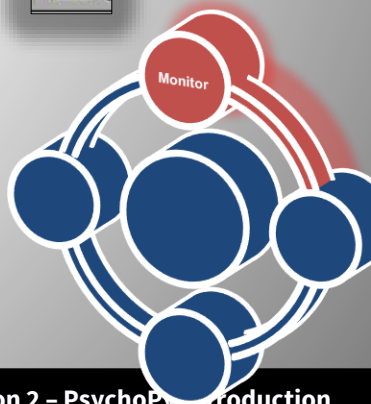
PsychoPy uses a monitor for presenting stimulus (text, images, sounds, videos); thus, how these objects are showed on the monitor (duration and way) is known as routines, therefore, all of the created routines represent the whole experiment itself. Within the routines, how the data (user answers) is collected should be set-up in order to save this information for a posterior analysis. 1234

INSTITUTO DE INFORMÁTICA UFRGS

PsychoPy
Psychology software in Python

Monitor Center

PsychoPy provides a simple and intuitive way for you to calibrate your monitor and provide other information about it and then import that information into your experiment.
Information is inserted in the Monitor Center (Tools menu), which allows you to store information about multiple monitors and keep track of multiple calibrations for the same monitor.
Also some attributes about the monitor can be set on Experiment Settings (next slides)



INSTITUTO DE INFORMÁTICA UFRGS

Monitor Center

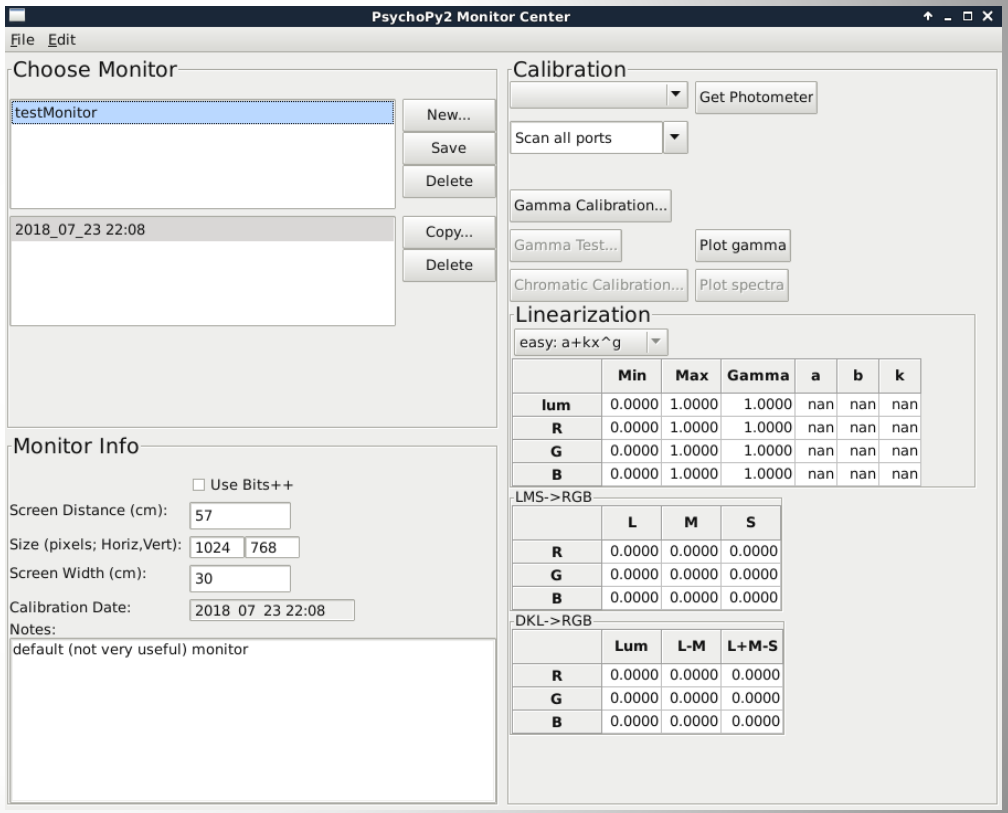
PsychoPy
Psychology software in Python

PsychoPy Coder

```
from psychopy import visual
win = visual.Window([1024,768], mon='SonyG500')
```

Class ***psychopy.visual.Window***(*size*=(800, 600), *pos*=None, *color*=(0, 0, 0), *colorSpace*='rgb', *rgb*=None, *dkl*=None, *lms*=None, *fullscr*=None, *allowGUI*=None, *monitor*=None, *bitsMode*=None, *winType*=None, *units*=None, *gamma*=None, *blendMode*='avg', *screen*=0, *viewScale*=None, *viewPos*=None, *viewOri*=0.0, *waitBlanking*=True, *allowStencil*=False, *multiSample*=False, *numSamples*=2, *stereo*=False, *name*='window1', *checkTiming*=True, *useFBO*=False, *useRetina*=True, *autoLog*=True, **args*, ***kwargs*)

PsychoPy Builder



.inf INSTITUTO DE INFORMÁTICA UFRGS

Experiment settings

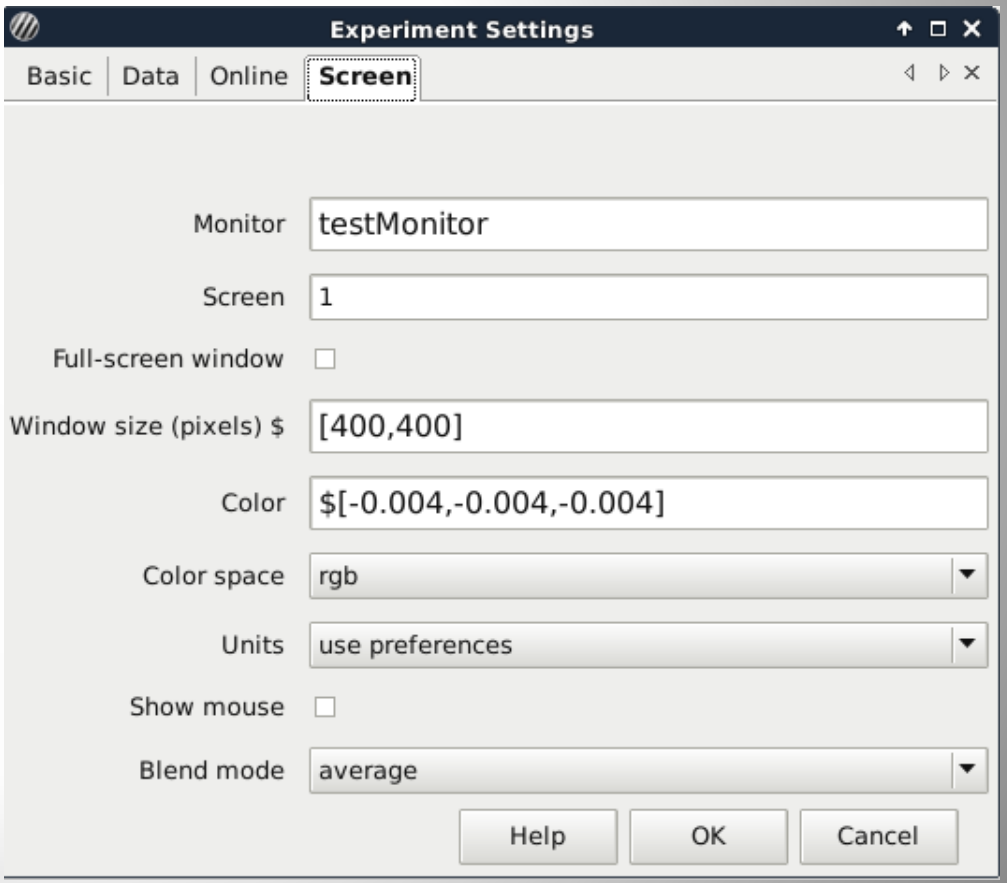
PsychoPy
Psychology software in Python

PsychoPy Coder

```
from psychopy import visual
win = visual.Window([1024,768], mon='SonyG500')
```

Class ***psychopy.visual.Window***(*size*=(800, 600), *pos*=None, *color*=(0, 0, 0), *colorSpace*='rgb', *rgb*=None, *dkl*=None, *lms*=None, *fullscr*=None, *allowGUI*=None, *monitor*=None, *bitsMode*=None, *winType*=None, *units*=None, *gamma*=None, *blendMode*='avg', *screen*=0, *viewScale*=None, *viewPos*=None, *viewOri*=0.0, *waitBlanking*=True, *allowStencil*=False, *multiSample*=False, *numSamples*=2, *stereo*=False, *name*='window1', *checkTiming*=True, *useFBO*=False, *useRetina*=True, *autoLog*=True, **args*, ***kwargs*)

PsychoPy Builder

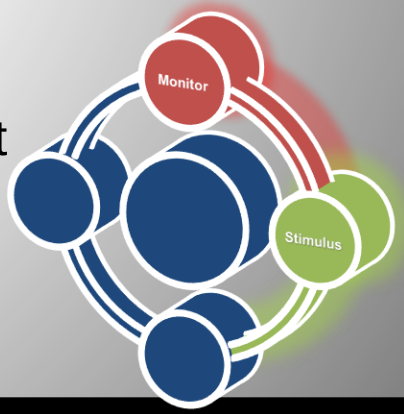


.inf INSTITUTO DE INFORMÁTICA UFRGS

PsychoPy
Psychology software in Python

Stimulus

- Aperture Component
- Cedrus Button Box Component
- Code Component
- Dots (RDK) Component
- Grating Component
- Image Component
- ioLab Systems buttonbox Component
- Keyboard Component
- Microphone Component
- Mouse Component
- Movie Component
- Parallel Port Out Component
- Patch (image) Component
- Polygon (shape) Component
- RatingScale Component
- Sound Component
- Static Component
- Text Component
- Variable Component

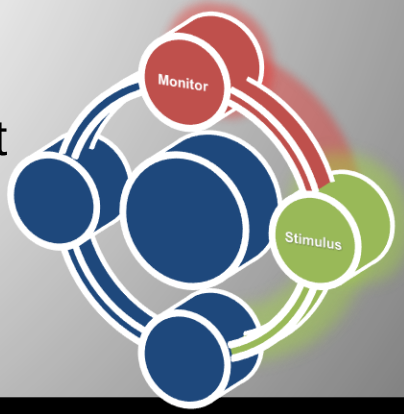


.inf INSTITUTO DE INFORMÁTICA UFRGS

PsychoPy
Psychology software in Python

Stimulus

- Aperture Component
- Cedrus Button Box Component
- Code Component
- Dots (RDK) Component
- Grating Component
- **Image Component**
- ioLab Systems buttonbox Component
- Keyboard Component
- Microphone Component
- Mouse Component
- Movie Component
- Parallel Port Out Component
- Patch (image) Component
- Polygon (shape) Component
- RatingScale Component
- **Sound Component**
- Static Component
- **Text Component**
- Variable Component

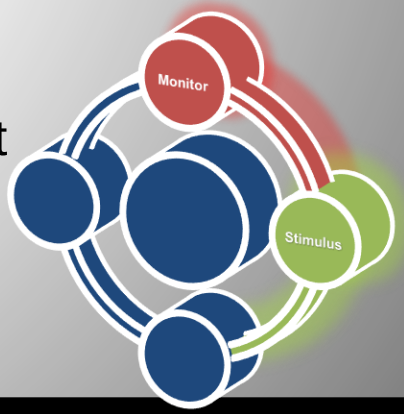


.inf INSTITUTO DE INFORMÁTICA UFRGS

PsychoPy
Psychology software in Python

Stimulus

- Aperture Component
- Cedrus Button Box Component
- Code Component
- Dots (RDK) Component
- Grating Component
- **Image Component**
- ioLab Systems buttonbox Component
- Keyboard Component
- Microphone Component
- Mouse Component
- Movie Component
- Parallel Port Out Component
- Patch (image) Component
- Polygon (shape) Component
- RatingScale Component
- **Sound Component**
- Static Component
- **Text Component**
- Variable Component



.inf INSTITUTO DE INFORMÁTICA UFRGS

PsychoPy
Psychology software in Python

Setting stimulus attributes

Stimulus attributes are features that each object has, for example, in a text, we have color, position, size… (Later)

Typically, they are set using either:

- A string, which is just some characters (a message.setText('world'))
- A scalar (a number)
- An x,y-pair (two numbers)

.inf
INSTITUTO DE INFORMÁTICA UFRGS

PsychoPy
Psychology software in Python

Setting stimulus attributes

Real world units

One of the particular features of PsychoPy is that you can specify the size and location of stimuli in units that are independent of your particular setup, such as degrees of visual. In order for this to be possible you need to inform PsychoPy of some characteristics of your monitor. Your choice of units determines the information you need to provide:

Units	Requires
'norm' (normalised to width/height)	n/a
'pix' (pixels)	Screen width in pixels
'cm' (centimeters on the screen)	Screen width in pixels and screen width in cm
'deg' (degrees of visual angle)	Screen width (pixels), screen width (cm) and distance (cm)

.inf INSTITUTO DE INFORMÁTICA UFRGS

PsychoPy
Psychology software in Python

Text stimuli

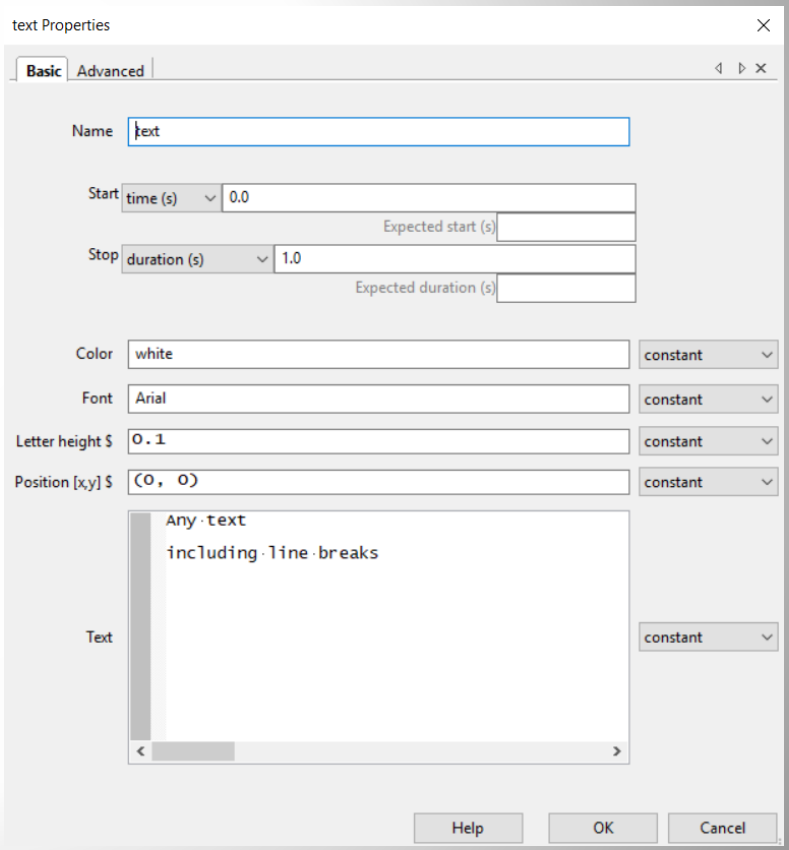
PsychoPy Coder

```
from psychopy import visual

message = visual.TextStim(win, text='hello')
```

class **psychopy.visual.TextStim**(*win*, *text*='Hello World', *font*='', *pos*=(0.0, 0.0), *depth*=0, *rgb*=None, *color*=(1.0, 1.0, 1.0), *colorSpace*='rgb', *opacity*=1.0, *contrast*=1.0, *units*='', *ori*=0.0, *height*=None, *antialias*=True, *bold*=False, *italic*=False, *alignHoriz*='center', *alignVert*='center', *fontFiles*=(), *wrapWidth*=None, *flipHoriz*=False, *flipVert*=False, *languageStyle*='LTR', *name*=None, *autoLog*=None)

PsychoPy Builder



INSTITUTO DE INFORMÁTICA UFRGS

http://www.psychopy.org/api/visual/textstim.html#psychopy.visual.TextStim

PsychoPy
Psychology software in Python

Image stimuli

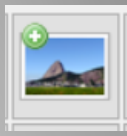
PsychoPy Coder

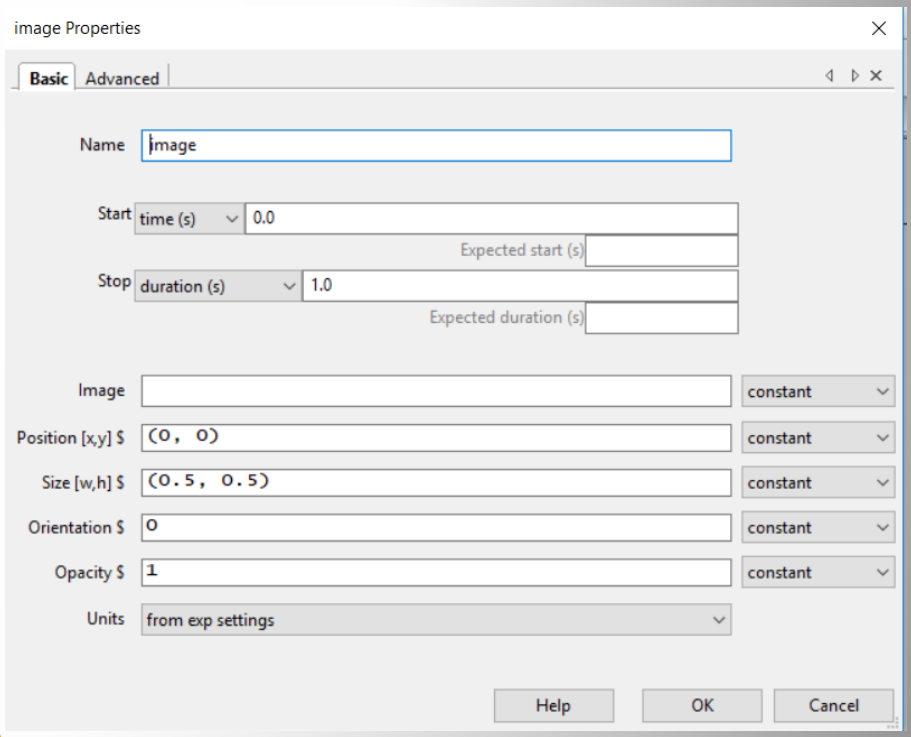
```
from psychopy import visual

img1 = visual.ImageStim(win=win,
                        image=filename,
                        units='pix',
                        size=(200,200))
```

class **psychopy.visual.ImageStim**(win, image=None, mask=None, units='', pos=(0.0, 0.0), size=None, ori=0.0, color=(1.0, 1.0, 1.0), colorSpace='rgb', contrast=1.0, opacity=1.0, depth=0, interpolate=False, flipHoriz=False, flipVert=False, texRes=128, name=None, autoLog=None, maskParams=None)

PsychoPy Builder





INSTITUTO DE INFORMÁTICA UFRGS

http://www.psychopy.org/api/visual/imagestim.html#psychopy.visual.ImageStim

PsychoPy
Psychology software in Python

Audio stimuli

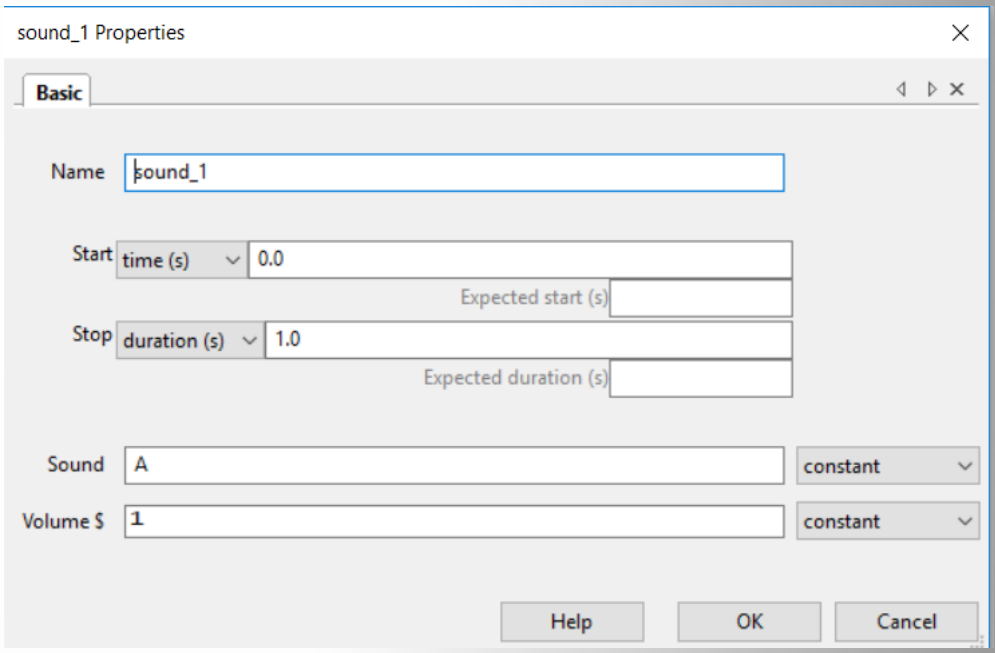
PsychoPy Coder

```
from psychopy import sound

aud1 = sound.Sound(value=soundfile,
                   volume=1)
```

class **psychopy.sound.backend_sounddevice.SoundDeviceSound**(value='C', secs=0.5, octave=4, stereo=-1, volume=1.0, loops=0, sampleRate=44100, blockSize=128, preBuffer=-1, hamming=True, startTime=0, stopTime=-1, name='', autoLog=True)

PsychoPy Builder



http://www.psychopy.org/api/sound.html

INSTITUTO DE INFORMÁTICA UFRGS

Our first PsychoPy code!!!

PsychoPy Builder

Let's do it our first window with a stimuli (text) using the builder.

- Please open PsychoPy (using the Virtual Machine);
- Set up the window to be used with a size of 400x400 and black as background color;
- Create a stimuli text with the word "Hello", color white and duration of two seconds;
- Create another stimuli text with the word "World", the same color and duration of two seconds. This must start after the previous one.

Lets go into hacking!!!

PsychoPy Coder

Let's do the previous example but this time using Python coding (Coder)

- Please open Spyder or any editor text (using the Virtual Machine);
- Call the necessary libraries (PsychoPy)
- Set up the window to be used with a size of 400x400 and black as background color;
- Create a stimuli text object with the word "Hello", color white;
- Set the draw operation;
- Call flip function (This really do the changes);
- Set the duration;
- Set the new value of the text object ("World");
- Repeat the same instructions after draw operation;

.inf INSTITUTO DE INFORMÁTICA UFRGS

Ok, Cool, but only text is pretty bored! Use some images!!!

PsychoPy Builder

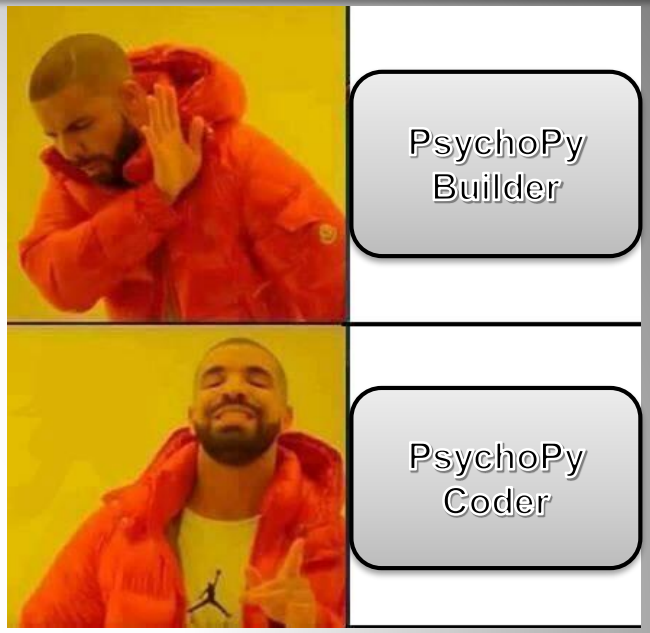
Let's do the previous example but this time using two images (Coder)

- Please open PsychoPy (using the Virtual Machine);
- Set up the window to be used with a size of 400x400 and white as background color;
- Create a stimuli image which will charge the file "hello.png", with a duration of two seconds;
- Create another stimuli image which will charge the file "world.png" and duration of two seconds. This must start after the previous one.

The files can be found in the /media folder.

.inf INSTITUTO DE INFORMÁTICA UFRGS

Ok Drake! Let's coding this example



- Please open Spyder or any editor text (using the Virtual Machine);
- Call the necessary libraries (PsychoPy)
- Set up the window to be used with a size of 400x400 and white as background color;
- Create an image object which has the file "hello.png";
- Create another image object which has the file "world.png";
- Repeat the same procedure as in the text stimuli.

The files can be found in the /media folder.

.inf INSTITUTO DE INFORMÁTICA UFRGS

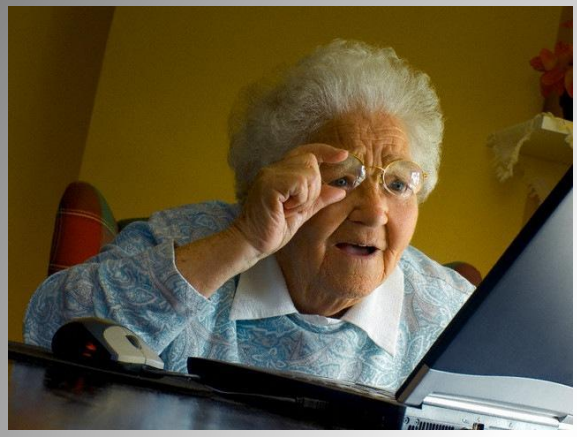
But… so for what do we show an image if actually, we can use an audio?

PsychoPy Builder

- Please open Spyder or any editor text (using the Virtual Machine);
- Call the necessary libraries (PsychoPy)
- Set up the window to be used with a size of 400x400 and gray as background color;
- Create an audio object which has the file “hello.wav” with a duration of one second;
- Create another audio object which has the file “world.wav” with a duration of one;

The files can be found in the /media folder.

.inf INSTITUTO DE INFORMÁTICA UFRGS



Coding again!!

PsychoPy Coder

- Please open Spyder or any editor text (using the Virtual Machine);
- Call the necessary libraries (PsychoPy)
- Set up the window to be used with a size of 400x400 and gray as background color;
- Create an audio object which has the file “hello.wav”;
- Create another audio object which has the file “world.wav”;
- Repeat the same procedure as in the image stimuli.

The files can be found in the /media folder.

.inf INSTITUTO DE INFORMÁTICA UFRGS

PsychoPy Coder

PsychoPy Builder

Let's do the previous examples all together!!!

- Please open both Builder and Spyder or any editor text (using the Virtual Machine);
- Set up the window to be used with a size of 800x800 and white as background color;
- Create an audio, image and text object with the "Hello" instance.
- The image must be placed at the centre and the text upper.
- All objects should appear together.
- The "world" objects (image, text and audio) should appear after.

.inf INSTITUTO DE INFORMÁTICA UFRGS

REMEMBER!!!! ALL SCRIPTS AND PROGRAMS ARE IN THE REPOSITORY

[1] http://psychopy.org/documentation.html

Any questions… ?