



CIGAL

CIGAL Workshop

Jim Voyvodic

May 13, 2015



Advanced topics

These CIGAL topics will not be covered in this session:

- Showplay's modular design
 - How to replace standard components with your own
- Structured event programming
 - How to let real-time events select stimulus lists
- Creating customized stimulus or response events
 - Programming your own real-time modules
- Adaptive paradigm programming
 - Adjusting task difficulty based on subject performance
- Real-time fMRI
 - Adapting stimuli based on brain BOLD response



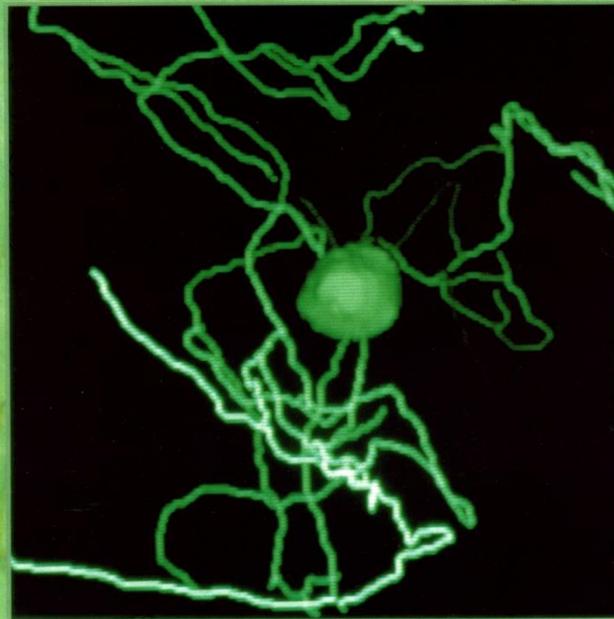
What is CIGAL?

- Compilable imaging, graphics, & analysis language
 - Similar in concept to Matlab
 - Originally written for video image processing and 3-D reconstruction graphics
- A single C program (CIGAL.exe) plus many text files
- Design goals:
 - Flexible language, Easy to use, Fast, Accurate
- Hardware interface abstracted via data variables
- Real-time multi-tasking processor added for fMRI

nature

INTERNATIONAL WEEKLY JOURNAL OF SCIENCE

Volume 336 No. 6195 10 November 1988 £1.95



TROPHIC REGULATION OF NEURON FORM

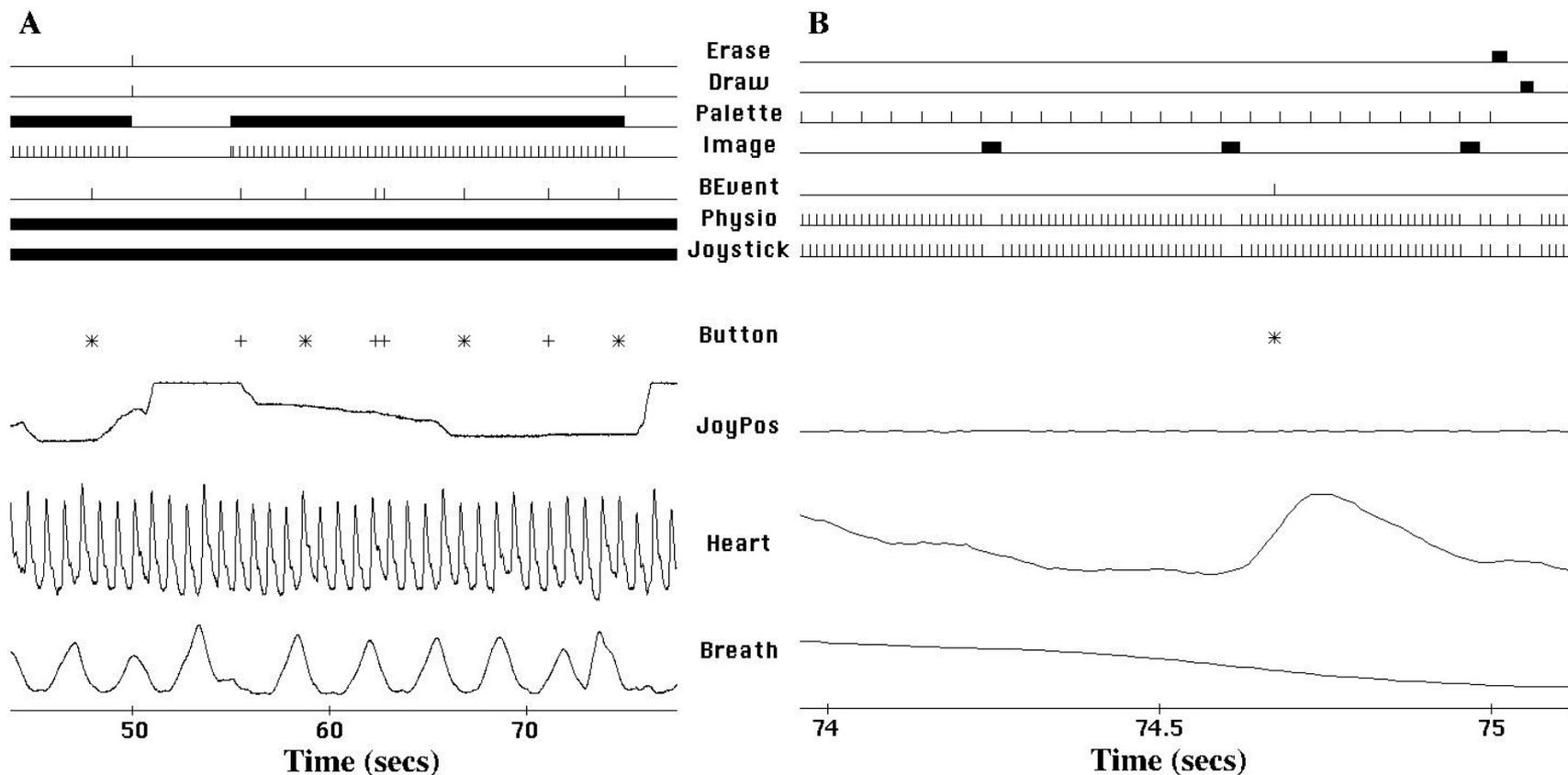
NEUROSCIENCES
product review

CIGAL has been used in neuroscience since 1985



Real-Time Multi-processing

What makes CIGAL special for fMRI studies is its real-time processor. CIGAL's real-time processor can run any number of multiple simultaneous program streams in parallel. Timing of events in each stream is automatically interleaved as necessary to ensure that every individual event occurs when specified. Actual execution timing is recorded to 20 us accuracy.





Showplay

Most fMRI paradigms are run using Showplay

Showplay is a CIGAL macro program ('showplay.imp')

Showplay calls other macro modules to read input files, prepare data tables, and save output files.

The task itself is a real-time program run within a single "realtime" command. Showplay automatically constructs the real-time program (rttime.tmp) at run time by linking together many real-time code modules. Different modules control independent real-time processing streams (e.g. scanner trigger, a/d read, stimuli, responses)

Advanced users can customize modules for their own needs.



Creating a Showplay Paradigm

Showplay was designed to provide a simple generic paradigm generation tool. The user sets a few parameters and then provides a list of stimulus events. Both the parameters and the stimulus list are usually specified together in a single plain text paradigm parameter file (PPF).

Stimulus events can indicate the exact time (onset and duration) and location (screen position) for each stimulus.

Showplay stimulus events can be (among others):

image (.jpg, .pcx)	rwait (wait for response)
sound (.wav)	owait (wait for operator)
movie (.avi)	reset (reset clock)
text (“”)	bdraw (draw a box)
fix (fixation cross)	edraw (draw an ellipse)
font (change text font)	writeport (write to a device)
quit	



Sample PPF Input File

```
showplay 0 ; initialize new showplay task
sbackcolor = "black" ; screen background color
stextcolor = 'white'
t_r = 2000 ; scanner TR
disdaqs = 3 ; initial discarded TR's
fslschedules = { ; for creating FSL analysis files (uses stim ID #s)
  fixation 1
  sounds 12 22
  faces 13 23
  movingdots 14 24
}
return ; ends parameter section
```

;Stimulus	ID	Flg	Onset	Duration	LocationXY	
"Press for faces"	99	0	0	4000	-1 -1	; show text
fix	1	0	4000	0	-1 -1	; central '+'
tones1.wav	12	0	6000	4000		
face1.pcx	13	1	8000	1000	200 100	
dots1.avi	14	0	12000	4000	-1 -1	
fix	1	0	16000	2000	-1 -1	
tones3.wav	22	0	-1	4000		
face2.pcx	23	1	-1	2000	400 200	
dots2.avi	24	0	-1	4000	-1 -1	
. . .						
quit	0	0	300000			



Running CIGAL

Start CIGAL using a StartCIGAL shortcut file located in the disk directory (folder) containing your task files. (Don't use the obsolete "start.cgl" method.)

If you don't have a startup shortcut, copy one from the CIGAL program directory containing the version you want to run.

At the BIAC scanners, you should use the shortcut:

LocalCIGAL

contained in the folder:

D:\Programs\CIGAL\CurrentVersion

(Running from the computer's local D: drive ensures that you use the appropriate default hardware settings and avoid unnecessary network traffic.)

On other BIAC computers, you can copy:

P:\CIGAL\CurrentVersion\StartCIGAL



Running a Showplay Paradigm

Once you have a PPF file, you can run it by selecting menu:

Showplay -> Load parameter file

then:

Showplay -> Run

Alternatively, you could type the commands:

```
showplay mypdigm.ppf      ; load file
```

```
showplay                  ; run
```

To end a paradigm (and save data), enter:

q or <Control>-E

To abort a paradigm (without saving data), enter:

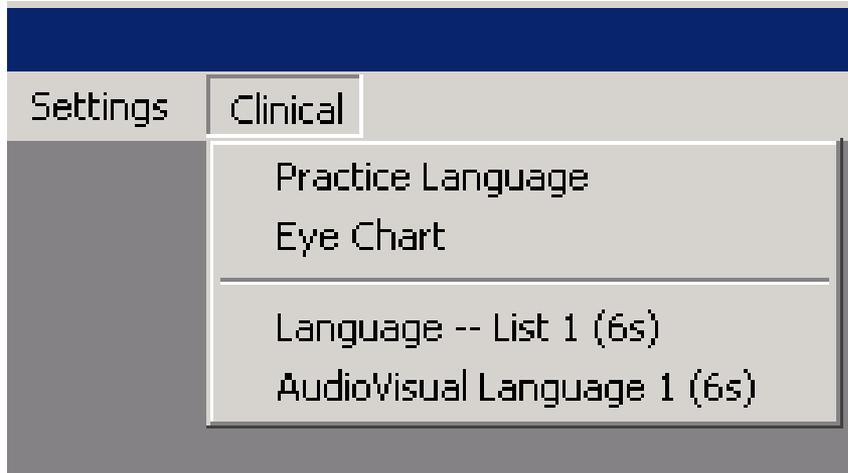
<Control>-C



Menus and Dialogs

All interactive menus and dialogs in CIGAL are simple plain text files (standard menus are stored in DLOGS in the CIGAL system folder – they can be used as samples).

The following is an example of a simple fMRI session menu:



To add a menu to your own session, simply create a file (eg “clinical.txt”) and then in your start.txt file put:
menu clinical.txt

```
"Practice Language
showplay 1 './PARAMS/prac_6s.ppf'
showplay

"Eye Chart
eyechart

"(-

"Language -- List 1 (6s)
Showplay './PARAMS/sent1.ppf'
showplay

"AudioVisual Language 1 (6s)
showplay './PARAMS/audvis6s_sent1.txt'
showplay
```



Response recording

The Showplay PPF typically only specifies stimulus events; multi-channel response monitoring is automatic, depending on available hardware devices and current parameter settings.

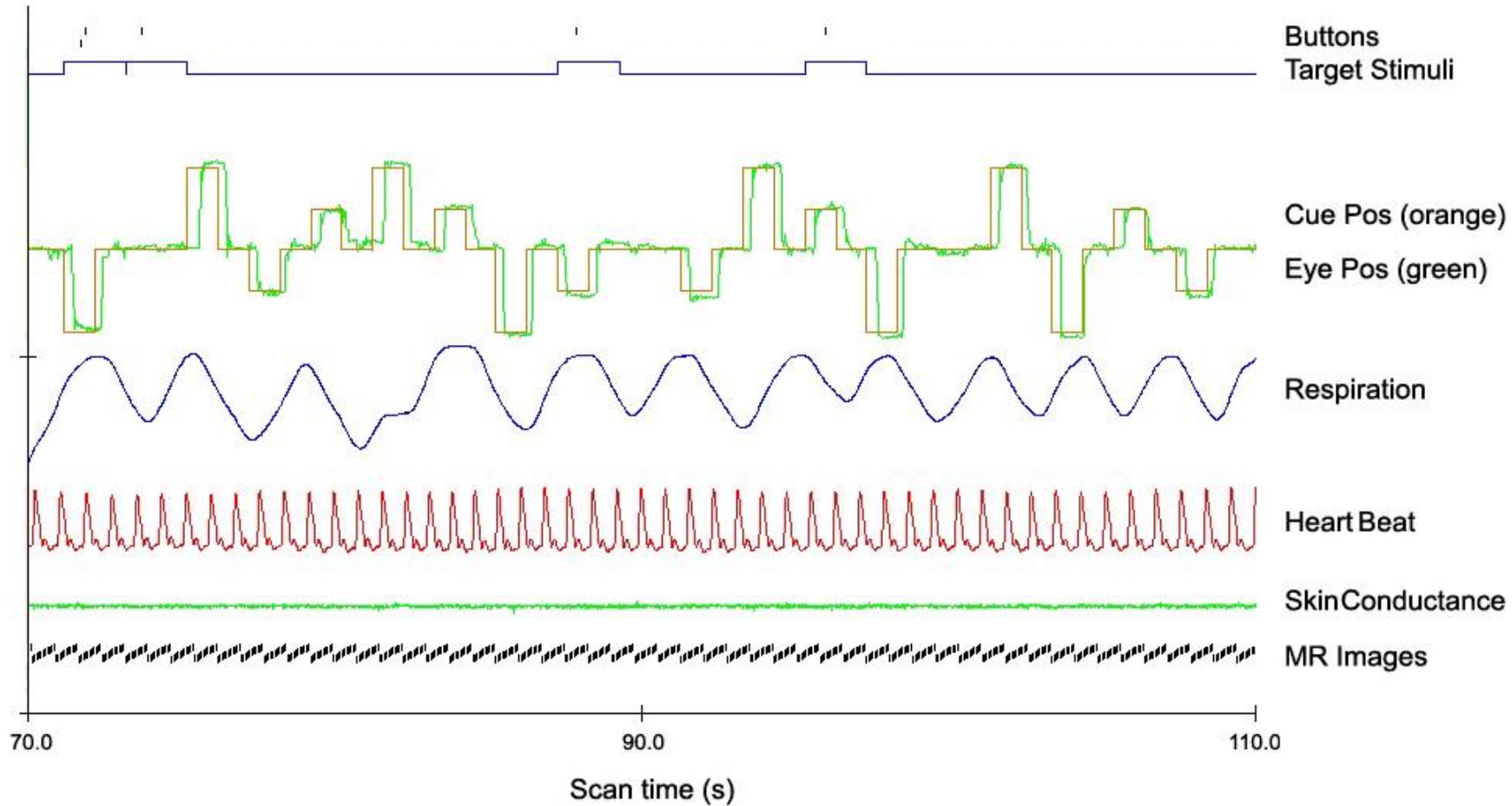
Usually, hardware devices connected to a computer are listed in a CIGAL configuration file (sysconfig or userconfig) for that computer and automatically initialized when CIGAL starts.

Device parameter settings are also typically controlled by a parameter file (eg. pdigmconfig or userconfig), or they can be specified explicitly in a start-up file (start.imp) or in a PPF file.

CIGAL's real-time modules poll all active input devices to automatically record events on keyboard, mouse, or any active button boxes. It also regularly samples (e.g. 100Hz) active devices that send continuous data (e.g. respiratory, Cardiac, GSR, eye-tracker, joystick).



Example Behavioral Data





Storing behavioral/physiological data

- CIGAL stores multiple types of data in 'pdigm' archive files (includes text and binary data)
 - Copy of task specification file and CIGAL scripts
 - All software and hardware parameter settings
 - Real-time run log (records all important events in chron. order)
 - Separate record for each recorded data channel
 - Copy of each reformatted output file
- Archive file is self-documenting
 - Internal header and optional XML metadata header
- CIGAL reformats data to accommodate analysis tools
 - <Events> XML format (stimulus and/or response events)
 - FSL "schedule" files
 - Performance scores
 - Physiological "time/value" text files



CIGAL Start-up Sequence

When you start CIGAL, the program looks for 3 start-up command files in your current folder and, if found, automatically runs the commands in each file.

The 3 files are searched for and run in the sequence:

To initialize external devices:

userconfig.imp (if found, used instead of sysconfig.imp)

To initialize your parameter preferences:

cigalsettings.imp (created by SAVE in settings dialogs)

To load menus and other start-up commands:

start.imp (or start.txt)



Download

All CIGAL software is distributed using the VSync (version synchronization) program. Instructions for downloading are available in the BIAC Wiki at:

<http://wiki.biac.duke.edu/jvs:cigal:download>

The download files themselves are available at:

<http://hawking.biac.duke.edu/CIGAL>

Username: download

Password:

(contact JV for password)

Download Zip file containing VSync and hooks for CIGAL:

VSyncCIGAL .zip

VsuncCIGAL_ReadMe.txt



Version control program - VSync

VSync is a stand-alone C program for synchronizing a local set of files with a remote set. VSync can be used on Linux, Windows, Unix, or Mac OS-X operating systems.

VSync keeps track of files grouped in 'projects'. A project can contain any type of file (e.g. text, data, binary programs). All files in a project are described in a project XML file. A master XML file is kept in a central repository. Running VSync locally downloads the remote XML and then generates a local XML table for the same files on your computer. It compares the two XML files to look for differences.

To download a VSync project you need to know the project name, the remote repository location, and you may need a user ID and password (or a decryption key). This information is usually automatically found by VSync by reading a local "vsync_project.xml" file.



VSync Features

- VSync is flexible and highly automated.
- The “Status” command informs you of any out-of-date files.
- “Download” will automatically update local files.
- All download changes are recorded in a local log file.
- All upload changes are recorded in a repository log file.
- You can selectively specify which files to download
- Old file versions remain on both the local computer (in VSync_bkup) and in the central repository.
- You can specify any earlier project version for download
- The central repository can be on a web-server or on a local file system (the default repository is at <http://hawking.biac.duke.edu>)
- If the project repository is a web-site URL you must have your computer connected to the internet to use VSync.
- VSync can be used locally as a simple file backup utility.
- By default, VSync simply checks version status. Use it often.



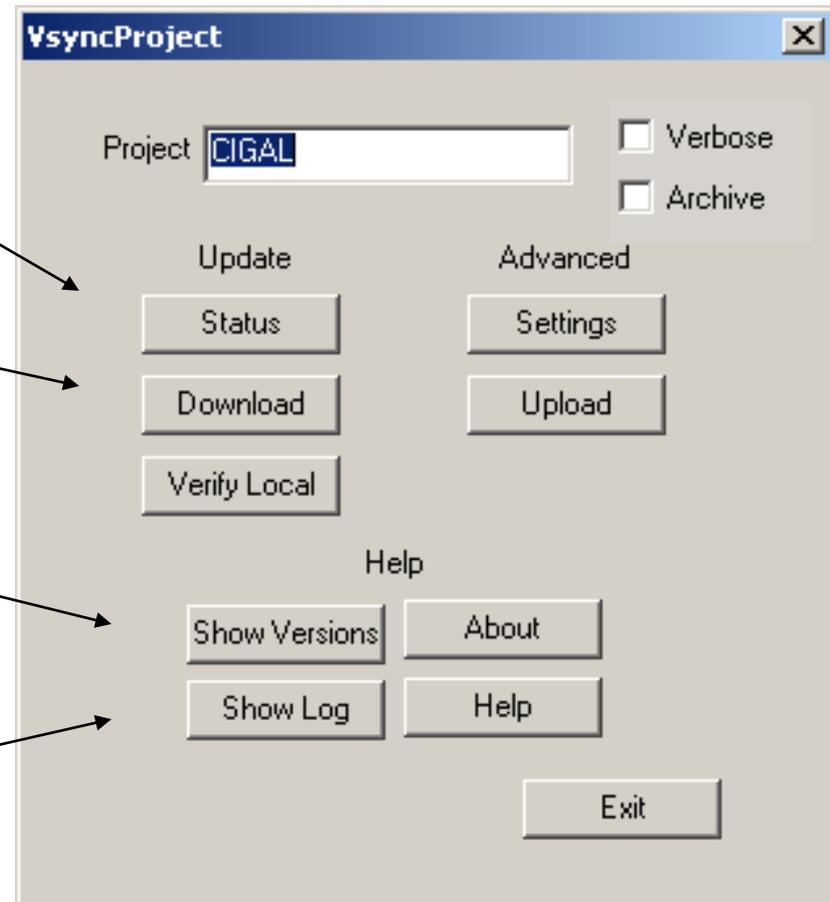
Running VSync on Windows

Status – compare to central copy

Download – Update local copy

Show versions – History of central changes

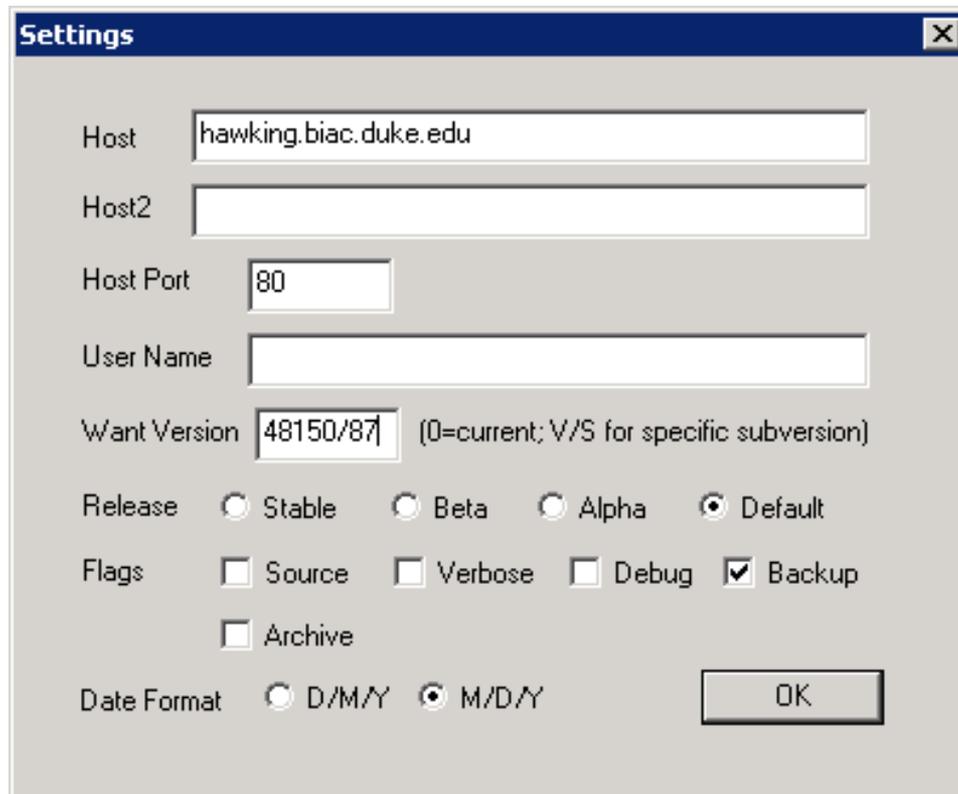
Show log – History of local changes





VSync Settings

You can use VSync to go back to any previous version of the project (see example below). The default (0) is the most recent version.



The screenshot shows a 'Settings' dialog box with the following fields and options:

- Host:
- Host2:
- Host Port:
- User Name:
- Want Version: (0=current; V/S for specific subversion)
- Release: Stable Beta Alpha Default
- Flags: Source Verbose Debug Backup Archive
- Date Format: D/M/Y M/D/Y
- OK button



Configuration Settings

The Settings menu can be used to check or change how CIGAL interacts with external devices.

Select “Configuration settings” to specify which devices to initialize.
Most useful available devices will be initialized by default.

Select “Hardware settings” to specify what information you want to record from the available devices (channels, sampling rates, etc).

Select “Scanner settings” to specify how to interact with the scanner.
Trigger options, default TR and Disdaqs, scan numbering, etc.

Changing and saving Configuration settings will create a file called:
userconfig.imp

Changing and saving Hardware or Scanner settings will create:
cigalsettings.imp



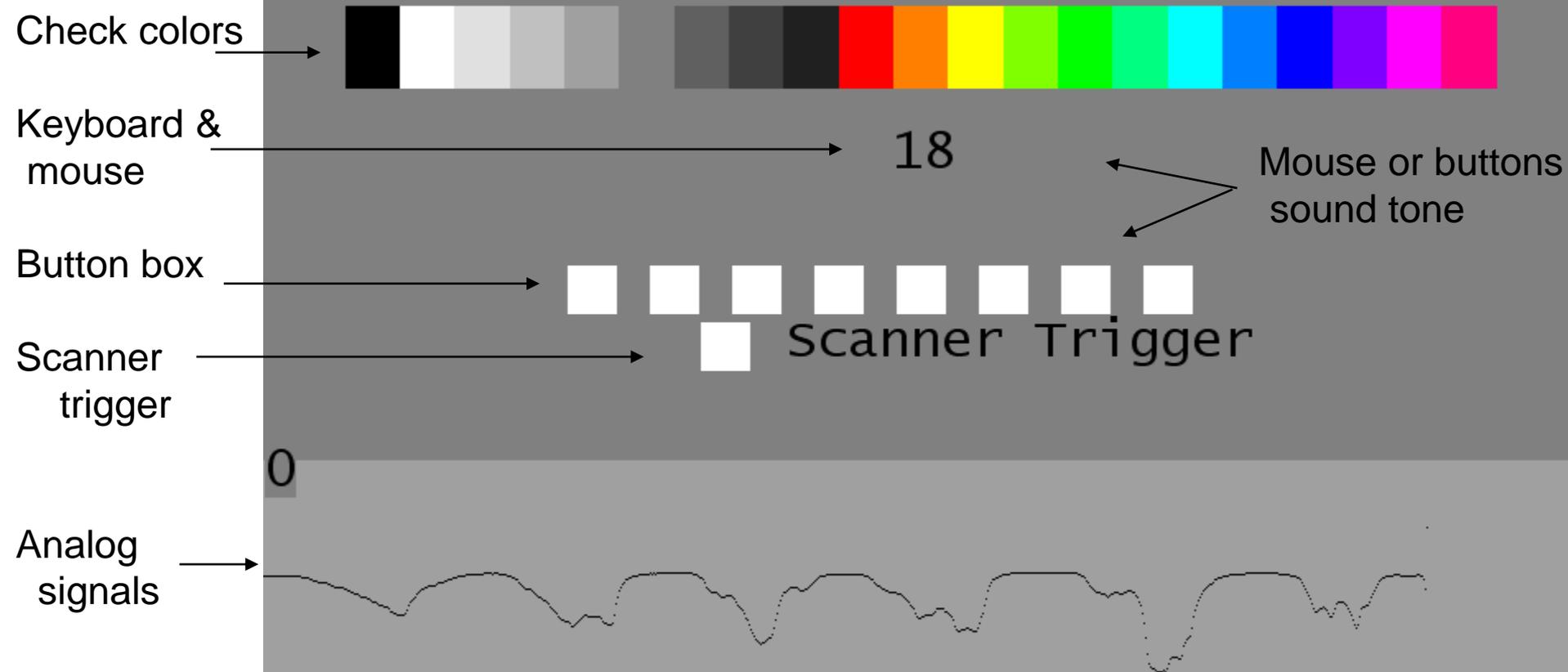
Quality Control

- Test I/O devices using menu option:
 - Showplay -> Test I/O devices
 - This tests video, sound, button boxes, scanner trigger, and analog input signals
- Test paradigms in fast-forward mode using:
 - Showplay -> Set double display speed
 - Doubles speed each time. Or set “showspeed” directly.
(for example: showspeed = 0.25 ; run at 1/4th – slow mo.)
 - Change “showspeed” between Load and Run.
- Real-time feedback on subject performance
 - Present audio or visual feedback for each response.
 - Can pre-code to report correct vs incorrect responses.
- View saved data immediately after paradigm
 - Check performance, physiology, and data logging



Test I/O devices

Test screen
<ESC> or <Ctrl>-C to finish



For analog: Channel=<PgUp>/<PgDn>, Offset=<Ins>/, Gain=<Home>/<End>



Documentation

On-line 'help' command

Wiki documentation:

wiki.biac.duke.edu/jvs:cigal

There is a tutorial page for using Showplay at:

wiki.biac.duke.edu/jvs:cigal:showplay:manual:tutorial

Wiki documentation can be continuously updated.



Known CIGAL Bugs

Screen blank at startup

Fix: Click mouse or press <Enter>

Menu click does nothing

Fix: Try again

CIGAL hangs at startup

Fix: Unplug and replug USB for MCC box

Funky paradigm colors via Remote Desktop

Fix: Don't run paradigms via Remote Desktop



Disclaimer

CIGAL, ShowPlay, and VSync were written by Jim Voyvodic.
They are the property of Jim Voyvodic and/or Duke University.

The software is made available for academic research use only, and is distributed “as is” with no warranty, guarantee, or promise of support.