Elisas Strange Case - Processing sketch

By f.Lüscher / fluescher.ch 2023 for Next Level Escape AG.

"AS IS" pi pa po etc.

Run this with processing.org or standalone when compiled on mac/win/linux/raspberry pi.

This was intended to run on raspberry pi 3.
Raspberry pi 4 does not work (yet) because: GPIO issues (related issue of processing) and autostart issues

When not on Raspberry Pi with GPIO pins and 4 connected rotary encoders,
set `GPIO_AVAILABLE` to `false` and `DEBUG` to `true`.

Press number keys `0`-`6` or left/right `arrow keys` to change stages manually.
Press `ESC` or `right mouse button` to go to desktop.

STAGES

Stage#	Content	Interaction	At end of stage..
0	Blackout.	Nothing works.	..waits for UDP signal
1	Message "AWAITING INPUT".	Flick the switch!	..waits for UDP signal
2	Startup sequence of computer.	Wait.	..**auto-jumps** to stage **3**
3	Elisas curves, without connected brainalizer on players head	Nothing works. Awaiting User to plug in Headset.	..waits for UDP signal
4	Elisas curves, with connected brainalizer.	Adjust with dials to sync brainwaves.	..**auto-jumps** to next stage when synched
5	Message "SUCCESS"	Wait.	..waits for UDP signal
6	Elisas thoughts as sequence in DE & EN	Wait.	..waits for UDP signal

UDP

Sending UDP Messages @ `53544`:

data	info
`sync_ready`	initially "loaded" stage `3`+`4` (only on startup)
`sync_boot`	is sent when user sflicks the switch on the computer
`sync_success`	both curves where properly aligned by the player

data	info
`sync_end_of_thoughts`	is sent after the last thought of elisa on stage `6`
`sync_died`	program closed or died

Listens to UDP Messages @ port `53545`:

data	info
`sync_stage0`, `sync_stage1` etc	Jump to a specific stage (`0`...`6`).
`sync_skipLoading`	can be used to skip the initial loading process if it takes forever. (Stage `3`+`4` will stay slow)
`sync_shutdown`	Gracefully shuts down the raspberry pi. Wait a minute to pull the power though.

IP & USER

The IP address is (maybe?) fixed to `192.168.86.68`.

- username raspberry pi: `esc`
- password raspberry pi: `synchron`

EXIT / RESTART APPLICATION

Press `ESC` or `right mouse button` to exit the program and see the desktop.
The program asks you to shut down.
Double click the file `play.sh` on the desktop to play the application.
Double click the file `update.sh` on the desktop to update the application.
To see the whole screen on one monitor, press the "SPLITTER" button on the "video wall hdmi" remote inside the computer case.
To reset the screens, press the "2x2" button on the remote.

UPDATE

If adjustments to the scripts are needed, call f.luescher 0787424834 or info@fluescher.ch.

After changes are made, double click the file `update.sh` on the desktop to pull latest changes made - be sure to deliver an internet connection (*disconnect Ethernet cable from back and make a wifi connection*). During loading, you'll see a new version number on the left screen pop up.

LUCKY NUMBERS

Knob	Target	from	to	Error margin
Amplitude	+**345**	327	363	±18
Frequency	+**307**	289	325	±18

Knob	Target	from	to	Error margin
Scale	+**12**	2	22	±10
De-noise	+**424**	374	474	±50

NERD STUFF

Deployment

1. To be safe, delete every `linux-*` deployment folder to make sure everything gets updated.
2. Build with processing 4 on mac (Processing 4 -> File -> export application -> Export). forget java. Build empties the folder(s) `/linux-*` first.
3. `git add .`, `git commit`, `git push` on mac
4. git pull on raspi 3

> *Note*:
> The `play_graceful_shutdown_arm.sh` &`play_graceful_shutdown_aarch64.sh` script auto-links the missing java library file into the compiled `/lib` folder because those files are not delivered by processing, see here.
>
> Each system must use its own:
>
> - For arm (rpi3):
>   `~/Applications/processing-4.1.2/modes/java/libraries/io/library/linux-armv6hf/libprocessing-io.so`
> - For aarch64 (rpi4):
>   `ln -s ~/Applications/processing-4.1.2/modes/java/libraries/io/library/linux-arm64/libprocessing-io.so`
>
> Those files need the be in the exported `linux-*/lib` folders.
>
> If those files are not present, the error will be `no processing-io in java.library.path`

Helpers

logging of boot:

```
tail -f ~/.cache/lxsession/LXDE-pi/run.log
```

start elisas_synchronotron:

```
sudo ~/Applications/sketchbook/elisas_synchronotron/linux-
arm/elisas_synchronotron
```

change startup things:

```
nano ~/.config/lxsession/LXDE-pi/autostart
```

If java is not found or java says "this application was build with a newer version of java":

EITHER: Update / install newest java

```
sudo apt install openjdk-17-jdk -y
```

OR: Use & make symlink to java that is used by processing editor (not needed if openjdk 17 is installed):

```
`sudo ln -s ~/Applications/processing-4.1.2/java/bin/java /usr/bin`
```

SETUP A NEW ELISAS SYNCHRONOTRON RASPI

Clone image from backup

- Use balenaEtcher to clone `24-04-23 Raspi3mitElisasSynchronisator.img.zip` from archive HDD for raspi 3.
- Use balenaEtcher to clone `24-04-23 Raspi4mitElisasSynchronisator.img.zip` from archive HDD for raspi 4.

> *Note*:
> Rpi4 is not yet ready because issue #1 and #2.

For a fresh install

1. Install raspian on an SD card

Use Raspberry pi Imager to flash an >=16GB micro SD card.

Set device, set Operating system (for raspi 4, choose Raspberry pi OS (64-BIT)), set target storage device, click "write".

2. Boot raspi first time

user: esc
pw: synchron

3. Get repository

Get the repo, it's public (now) so don't worry about nothing.

```
cd ~/.
mkdir Applications && cd Applications
mkdir sketchbook && cd sketchbook
git clone https://github.com/falue/elisas_synchronotron.git
```

4. Configure autostart

- ARM: Copy `scripts/autostart_arm` to `~/.config/lxsession/LXDE-pi/autostart` (rename to just `autostart`)
- AARCH64: Copy `scripts/autostart_aarch64` to `~/.config/pcmanfm/LXDE-pi/autostart` (rename to just `autostart`) *Note*: Not yet tested

5. Install processing 4.1.2 for raspian

Get this version: https://github.com/benfry/processing4/releases?q=4.1.2

```
tar -xvzf processing-4.1.2-linux-arm64.tgz
```

- move to `Applications/processing-4.1.2`
- Try to start the application and open the main .pde file. Start the animation. Does it run?

Library problems:

- No library found for controlP5 -> in library manager look for "**controlP5**"
- No library found for processing.io -> in library manager look for "**processing-io**" -> "processing.io" -> "Hardware I/O"
- No library found for hypermedia.net -> in library manager look for "**UDP**" by Stephane Cousot (!!)

5. Install java (maybe?)

```
sudo apt install openjdk-17-jdk -y
```

6. Move fitting files to desktop

Copy `update.sh` + `play.sh` shell script files from `scripts/` to desktop for easy access.

7. Allow files to be executed directly

Folder window -> Edit -> Preferences -> check "Dont ask options on launch of executable file"

8. Set background image

Find something fancy.

Enable SSH

Start Menu -> Raspberry Pi Configuration -> Interfaces -> Check SSH
Because why not?

Enable GPIO

Start Menu -> Raspberry Pi Configuration -> Interfaces -> Check REMOTE GPIO
(potentially useful)

10. Other stuff for screen resolution etc?

Enjoy!