

# 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](https://processing.org) or standalone when compiled on mac/win/linux/raspberry pi.

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.	<b>..auto-jumps</b> to stage <b>3</b>
3	Elisas curves, without connected brainalyzer on players head	Nothing works. Awaiting User to plug in Headset.	..waits for UDP signal
4	Elisas curves, with connected brainalyzer.	Adjust with dials to sync brainwaves.	<b>..auto-jumps</b> 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
<code>sync_ready</code>	initially "loaded" stage <code>3+4</code> (only on startup)
<code>sync_success</code>	both curves where properly aligned by the player
<code>sync_end_of_thoughts</code>	is sent after the last thought of elisa on stage <code>6</code>
<code>sync_died</code>	program closed or died

Listens to UDP Messages @ port `53545`:

data	info
<code>sync_stage0</code> , <code>sync_stage1</code> etc	Jump to a specific stage (0...6).
<code>sync_skipLoading</code>	can be used to skip the initial loading process if it takes forever. (Stage 3+4 will stay slow)

## IP & USER

The IP address is fixed to **192.168.178.97**.

- 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.

Double click the file **play.sh** on the desktop to restart 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\_and\_play.sh** on the desktop to pull latest changes made - be sure to deliver an internet connection. During loading, you'll see a new version number.

## LUCKY NUMBERS

Knob	Target	from	to	Error margin
Amplitude	<b>+345</b>	327	363	±18
Frequency	<b>+307</b>	289	325	±18
Scale	<b>+12</b>	2	22	±10
De-noise	<b>+424</b>	374	474	±50

## NERD STUFF

### deployment

1. **move** the file **libprocessing-io.so** from **/linux-arm/lib** out of the ways before deployment

2. Delete folder `/linux-arm/lib` because sometimes Processing does not deploy the newest version
3. Build with processing 4 on mac. forget java. Build empties the folder `/linux-arm` first.
4. **move** the file `libprocessing-io.so` to `/linux-arm/lib` again. It is also available in the `_tools` folder.
5. git add, git push on mac
6. git pull on raspi

logging of boot:

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

start elisas\_synchronotron:

```
sudo /home/esc/Applications/sketchbook/elisas_synchronotron/linux-arm/elisas_synchronotron
```

change startup things:

```
nano /home/esc/.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 /home/esc/Applications/processing-4.1.2/java/bin/java /usr/bin
```

If something with "libprocessing-io not found":

---

## Use & make symlink to missing native io library (if )

```
ln -s ~/Applications/processing-  
4.1.2/modes/java/libraries/io/library/linux-armv6hf/libprocessing-io.so  
lib/
```