Computer responses to key presses

Carlos Antonio Rodríguez-Saltos

December 19, 2018

Here I documented the development of the code that specified the responses of the Raspberry Pi to the press of a key. This document forms part of a series of documents documenting the development of the SingSparrow operant-yoked mode. For more information on SingSparrow and on the operant-yoked mode, see …

Simulating presses

I needed to simulate a generator of key presses and a scanner of presses. Both processes had to be separate modules, but a bridge had to be set from the generator to the scanner. In this section, I will describe the generator.

In ultimate instance, the generator must be the user (or bird) interacting with the keys connected to the GPIOs of the RPi. But in an initial phase of testing, the generator was me pressing keys on the keyboard of my computer.

I wrote a simple `bash` script to simulate key presses. The script reads key strokes on the keyboard and writes them to a text file that then the scanner function will read In the text file, there are two columns, one for each key. Two keys on the keyboard are selected as the "left" and "right" key equivalents of the RPi set-up. When the user is not pressing those keys, then the generator function writes zeros; if the keys are pressed, it writes ones. To prevent having a large file with rows of zeros, the generator function allows only two rows on the text file at a time; it constantly updates the rows based on new user input.

The name of the generator script is `press-capture.sh` (it is included in the source folder of this document).

Detecting a press

The Raspberry Pi (RPi) will be scanning contantly for key presses. To detect the onset of a press, the RPi will compare the state of each key to their previous state; a state of 0 (not being pressed) succeeded by a state of 1 (press) is the definition of the onset of a press. To simulate the information needed to detect presses, I made a 2X2 matrix. The first row contains information on the state of the keys and the columns corrspond to the keys themselves.

```
keys = [[0,0],[0,0]]
print(keys)
```

```
## [[0, 0], [0, 0]]
```