CirclesGame documentation:

The aim for this document is to provide some documentation for the setup phase for the app.

The app/game was developed as a tool for a research based on "Individual attachment style modulates human amygdala and striatum activation during social appraisal", which can be found in the link below[1].

The app written in JavaFX and consists of 3 parts:

- the game logic itself
- a data handler which gather the data from the game and saves it into ./data/circles_game.csv
- an additional data handler which sends signal over a TCP connection during certain points in the game to an OpenVibe acquisition server

Config:

The app can be configured to some extent. All app related configuration values are located in src/main/resources/config.json

Note that you can change the names of each column in the .csv file, however you can't change what data will be gathered. If you add a column to the list/remove one - it'll not affect the actual data you'll get.

The images can be changed too. To change them - make sure you place them in src/main/resources/images, and change the values in config.json under "images" to:

"image/<your_image_name>.<image_extention>": "<feedback_type>"
(<feedback_type> can be either "positive" or "negative", any other strings will break the app and will cause it to crash)

Compilation:

- clone the source code[2]
- import the project as a Maven project

[1] https://www.researchgate.net/publication/23151714_Vrticka_P_Andersson_F_Grandjean_D_Sander_D_Vuilleumier_P_Individual_attachment_style_modulates_human_amygdala_and_striatum_activation_during_social_appraisal_PLoS_ONE_3_e2868

[2] git clone https://github.com/AvihaiAdler/CirclesGame.git

- compile by running mvn compile assembly:single in your terminal

- you can run the app with java -jar CirclesGame.jar (which you can find in the target/ directory)

OpenVibe:

To quote OpenVibe themselves "OpenViBE is a software platform that enables to design, test and use Brain-Computer Interfaces (BCIs). OpenViBE can also

be used as a generic realtime EEG acquisition, processing and visualization system."[3]

OpenVibe consists of 2 parts: acquisition server, and a Designer. The server handles the data received from external applications through a TCP connection and send it to the Designer. The Designer process the data using various 'boxes'.

OpenVibe requires python to work, make sure you have python installed.

To set up OpenVibe to work with the app do the following:

Acquisition Server:

- open an acquisition server instance

- Driver: if you have EEG connected choose it, otherwise, for testing purposes a Generic Oscillator should be fine

- Connection port: choose a port, the port number must match the port in the Designer

- Sample count per set block: set to 32

- Preferences: the port the server uses to communicate with our app is under TCP_Tagging_Port. Make sure it matches the port in the app's config.json file

- press Connect to initiate the server

- press Play to start send/receive data

Designer:

- open a Designer

[3] http://openvibe.inria.fr/faq/

- drag & drop an 'Acquisition Client' box

- make sure it listens to the port the Acquisition Server is set to, you can check the port by double clicking on the box

- drag & drop a 'Signal Display' box

- connect the 2 boxes by drawing the lines between the Client 'Signal' output to the Signal Display 'Signal' input, and between the Client 'Stimulations' output to the Signal Display 'Stimulations' input

Congratulations! the app is now set up and ready to go