Pippin Barr > CART263 > Project Proposal > illiez > Telomatic > 22/03/22

Artistic vision

Telomatic is an exploration of post-digital identity and intersubjectivity through computational electronic art. It consists of a webpage proposing a ludic drawing activity while surreptitiously transmitting user interaction data.

Drawing inspiration from telematic art, technology ethics and cypherpunk ideas, Telomatic seeks to comment on and propose playfully engaging alternatives to the dematerialization of contemporary virtualized landscapes, offering an interactive experience reverse-engineering embodied possibilities, as well as address concerns of cybersecurity in our increasingly connected age.

The viewer engages with the piece through an apparently innocuous web application proposing an augmented reality drawing activity using machine-learning trained models allowing real-time hand pose detection. Following the position of the index finger, the user input traces a line of which the weight and colour can be modulated. The data generated by this activity is also wirelessly transmitted to a Bluetooth Low Energy [BLE] receiver.

While outside the scope of the current proposal, it can be mentioned that in the current configuration, the data is received by a microcontroller which actuates a motor embedded in an ad hoc silicon housing. Previous iterations have harnessed other technological modalities, such as a video-conferencing application and a smartphone application, as data sources.

The proposal here is to develop a webpage which would capture a video feed, run it through a hand position recognition model to discern the number of currently active hands, their handedness and the position of their indexes, draw a line following the tips of right index fingers, permit left-handed interaction with line style aspects and send data via BLE. The page would be coded using HTML, CSS and Javascript, and would implement the p5.js, ml5.js/tensorflow.js and p5.ble.js libraries.

Technical challenges

The main technical axes are hand recognition, wireless communication and webpage styling.

For **hand-position recognition**, we have already been exposed through lectures to the ml5.js library, which contains a Handpose feature. The project prototype has been developed from this example, extracting the index tip finger marker to draw a graphic element on the screen. However, Handpose presents a certain number of limitations of which – most crucially for the present purposes – processing load and single-hand restriction. The final project will attempt to address these issues through the implementation of an alternative library (the replacement of MediaPipe's Handpose library vulgarized by ml5.js, the more recent Hand Pose Detection). While I anticipate this to be a non-negligible technical challenge, requiring a deeper dive in machine-learning model implementation than previously used libraries, the advantages of multi-hand detection, chiral distinction and decreased computational load make the attempt worthwhile.

With respect to **wireless communication**, this will require the implementation of the p5.ble.js library (the choice of BLE over other forms of wireless communication are outside the scope of this proposal and will therefore not be addressed presently). Due to previous failed experiments attempting to implement BLE connectivity in computational settings (Max/MSP, Android Studio, Processing) with a steep learning curve and scanty success, I expected this to be the most challenging and jeopardizing aspect of the project. However, initial forays – such as those undergirding the prototype – have been encouraging. Much remains to be done in terms of the connectivity (e.g. streamlining pairing with specific target, automatizing reconnection, etc..) and communication (e.g. further pre-digesting/simplifying message sent over BLE, stop/clear upon disconnect), but the task appears less daunting/perilous.

Finally, while apparently the simplest task, **web page styling** is shaping up to be the most sanity and time consuming technical challenge. Concretely, this aspect encompasses the addition of UI buttons and sliders within the canvas serving as a holder for a video feed and graphic element. Multiple attempts have been and will continue to be made to create and style these UI features, including basic HTML/CSS, jQueryUI widgets and various Javascript/p5.js libraries (p5.clickable, p5.gui). As user-developed libraries, the latter have evinced limitations such as imperfect function or incomplete features, and are less straightforward to implement than the former two. However, while more rapidly implementable and stable, HTML/CSS and jQueryUI remain difficult to integrate within the p5.js framework, especially in the context of responsive design. This technical challenge has led to a crossroads, with the choice between relinquishing the idea of a responsive canvas and hard-coding the UI features using a blend of HTML, CSS and jQueryUI or further exploring p5.js libraries (e.g. p5.touchgui), which might be the wiser choice given the downstream desire for a causal effect of hand-pose recognition on UI features (i.e. entirely dematerialized interaction with webpage). Both shall be explored.

Another feature up for inclusion is saving and/or transmitting the graphic element upon completion by the user. This could take the form of a printed page, e-mail or QR code. TBD.

Pippin Barr > CART263 > Project Proposal > illiez > Telomatic > 22/03/22

Visual sketches

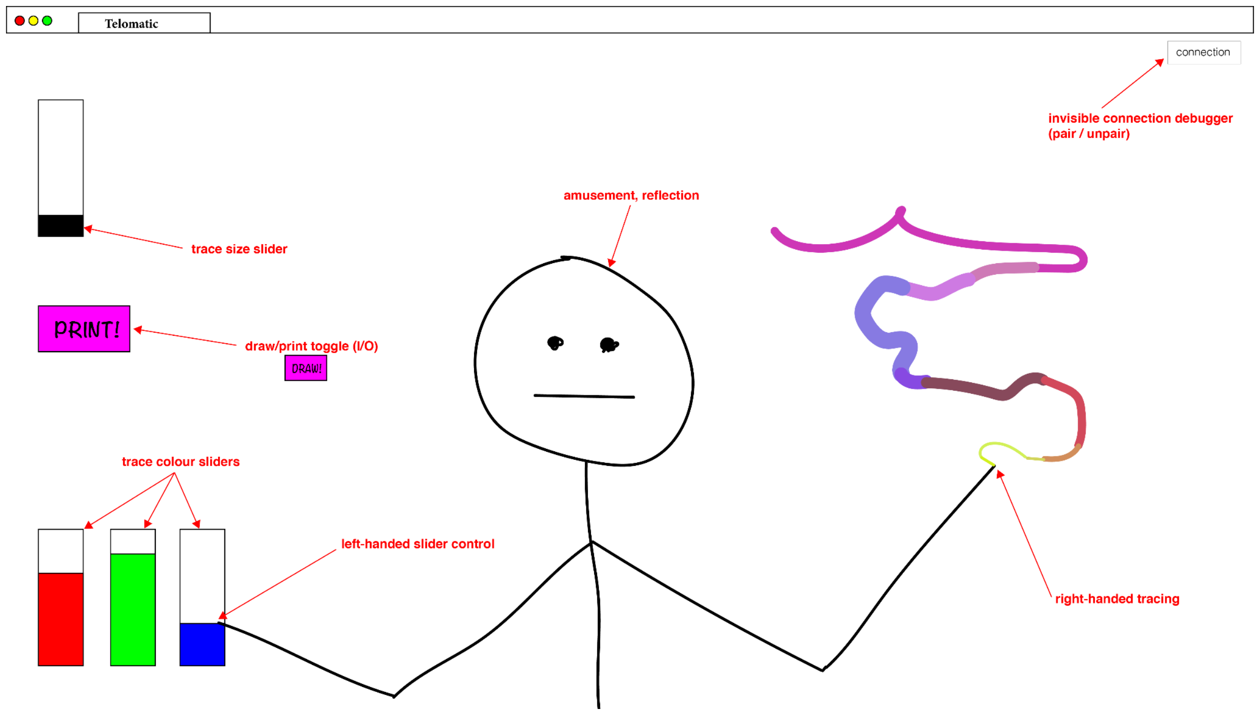

Figure 1: Mock-up of webpage appearance

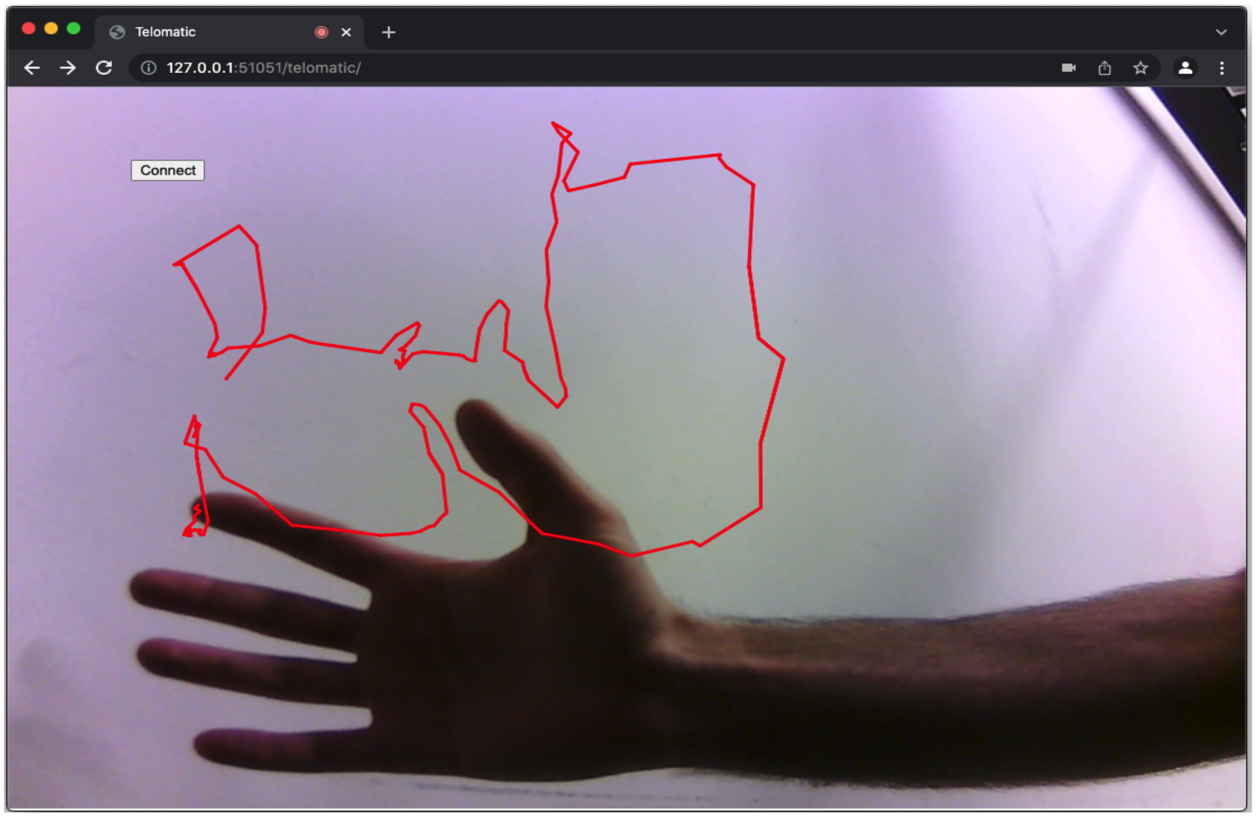

Figure 2: Screenshot of prototype in use