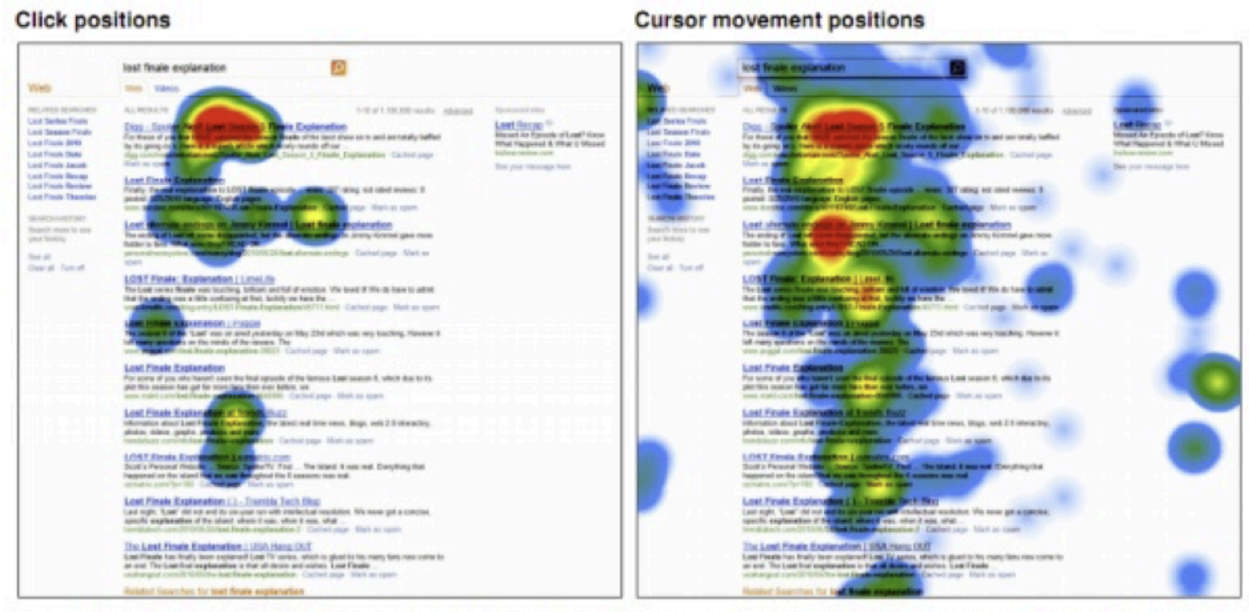
HCI 5840 Project Spec
Edward Cody
GitHub: https://github.com/Edward-Cody/EdwardCody_Project

General description of the project

The tool will be a web browser plug-in referred to as Cursor Tracker (for now). It will track the movements of a user's cursor and the clicks, and be able to visualize the collected data in a GUI. Website developers and individuals testing websites would be the users. The developers will install and use Cursor Tracker to better understand how users navigate their websites. Currently, at my job, no data on cursor movement or clicks is gathered. This data could reveal issues we are unaware of and help us design better websites. For example, perhaps users don't click on an interactive element of the website because they don't realize they can.

Cursor Tracker will continuously gather data on the cursor position of the user while the plug-in is installed. A feature potentially added later in development would allow the user to specify what websites Cursor Tracker records data from. The user would perodically access the Cursor Tracker GUI to see a report of the data gathered. The recorded data is converted into a heatmap (see the image below) using [visual-heatmap](https://github.com/nswamy14/visual-heatmap) (https://github.com/nswamy14/visual-heatmap).

The heatmap results are visible in the GUI (a web app using Flask). The heat map will have a level of transparency with the analyzed website underneath. The website underneath will either be the actual live website or a screenshot. Visualization of other data, such as time spent on a webpage, will be explored after the heatmaps are functional. This GIU could use Streamlit to display the gathered data in alternate ways. It is possible that the GUI aspect of Cursor Tracker could be run with a command line interface, but this is not preferred.

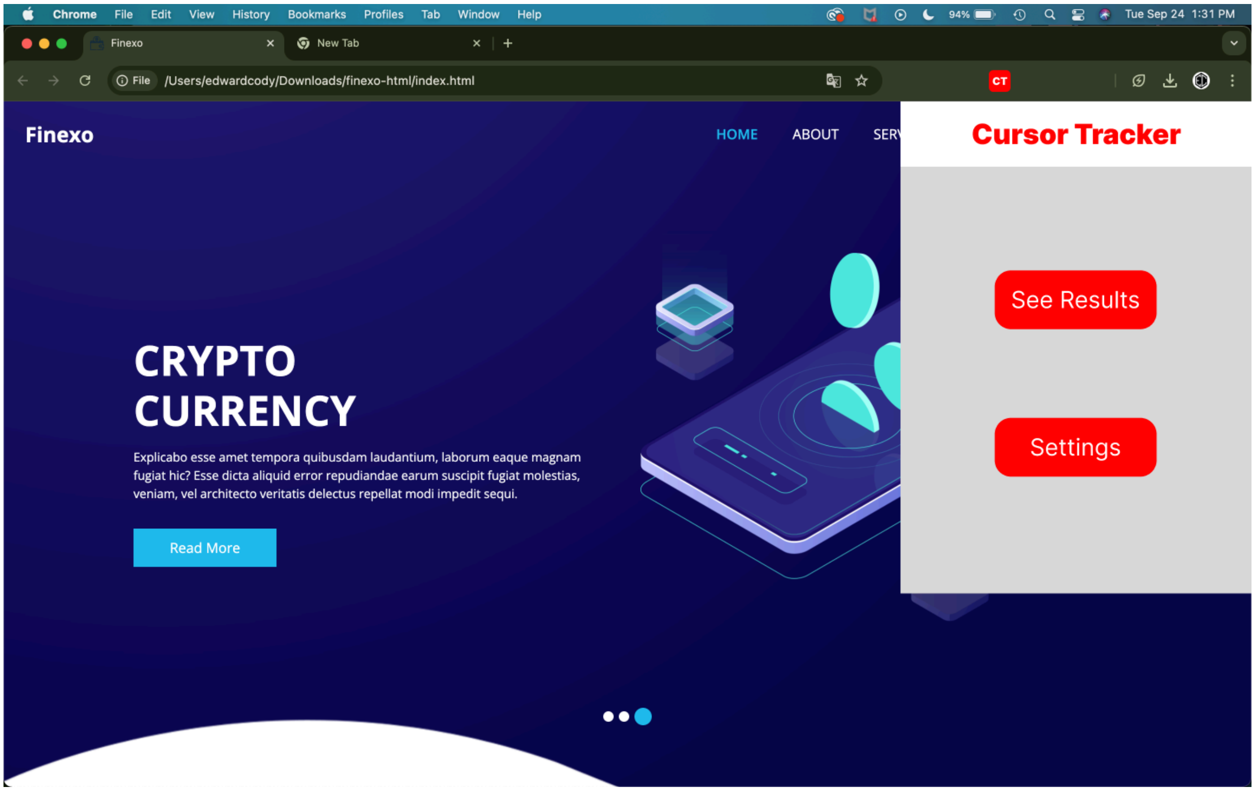

Click positions
Cursor movement positions

Task Vignettes (User activity "flow")

A web developer creates a new web app that includes a map with pins on it, which can be clicked to reveal more information. The developer is curious if the interface is intuitive and sends it to 20 people to test. The Cursor Tracker plug-in is installed in the tester's browsers. After the web app has experienced some user testing, the developer opens the Cursor Tracker GUI. From there he opens the cursor movement heatmap of his web app to see how the users are navigating and interacting with his web app. The developer can see that there is little cursor movement over the pins on the map. So, he switches from the cursor movement heatmap to the click heatmap. He notices that the pins on the map are rarely clicked, and realizes that it is not obvious that the pins are clickable. The developer now makes changes to his web app based on the results of the testing that he could see with Cursor Tracker.

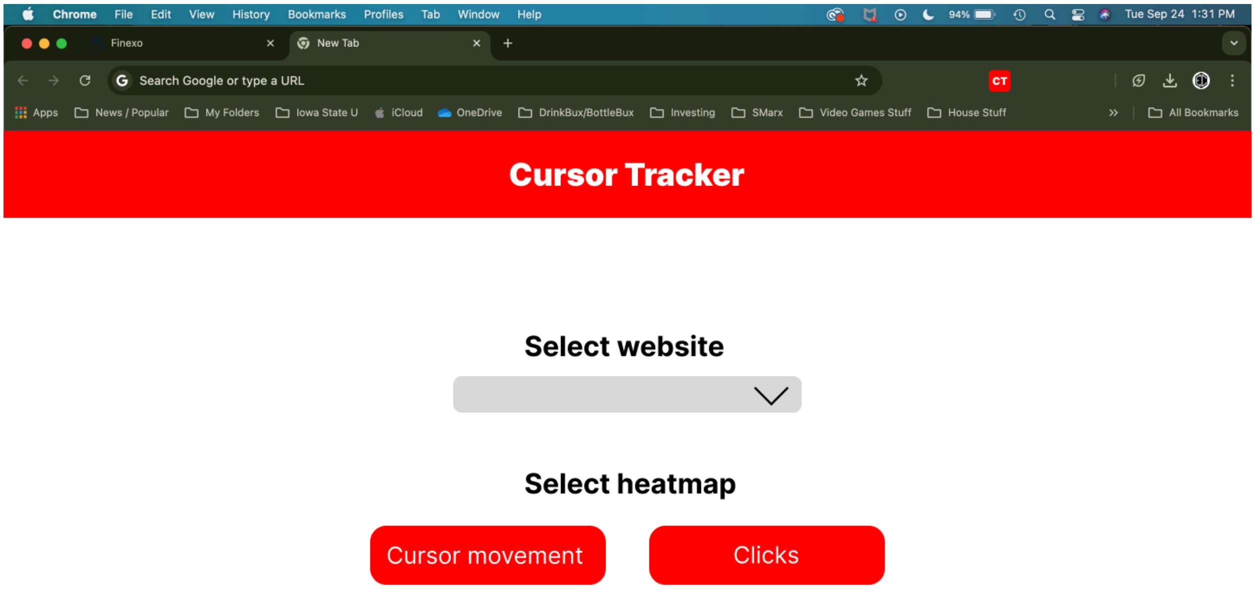
- The Cursor Tacker is already installed in the browsers the people performing the testing use.
- The movement and click data are continuously recorded and stored in a JSON file. For version 1, this data is saved locally. Future versions can look into storing this in a database.
- The gathered data will record the size of the browser window in pixels (if possible).

1. Cursor Tracker overlay open

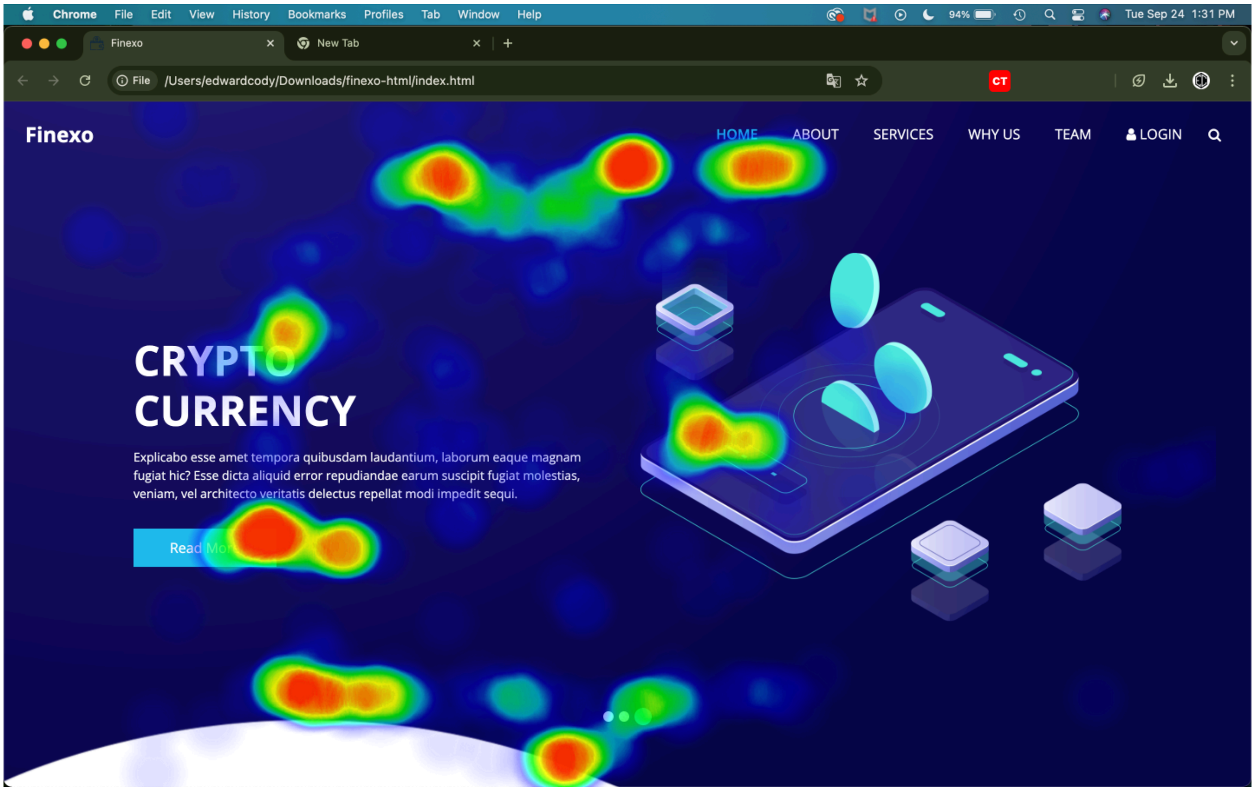


Explicabo esse amet tempora quibusdam laudantium, laborum eaque magnam fugiat hic? Esse dicta aliquid error repudiandae earum suscipit fugiat molestias, veniam, vel architecto veritatis delectus repellat modi impedit sequi.

2. Cursor Tracker GUI

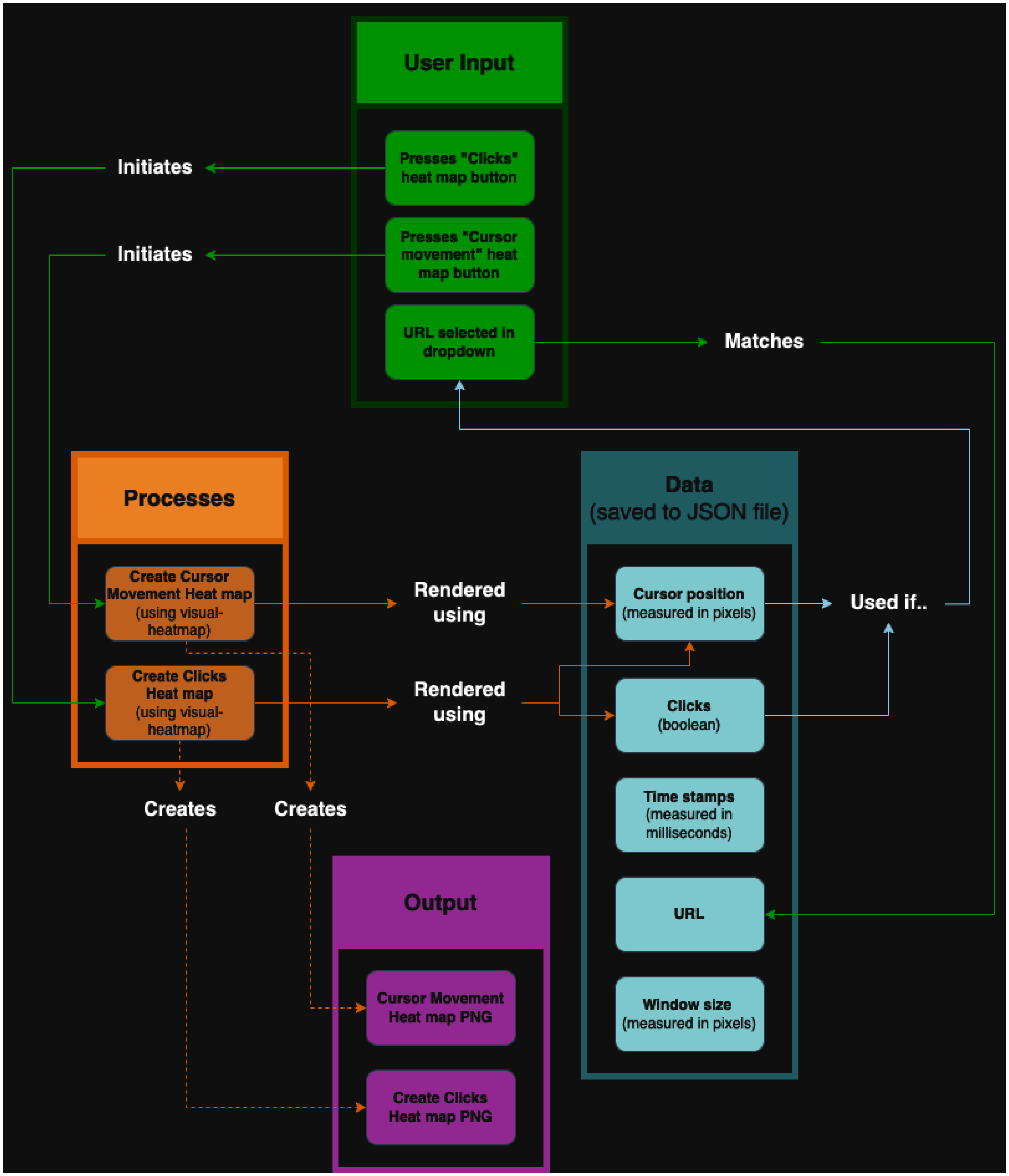


3. View of a heat map over the website

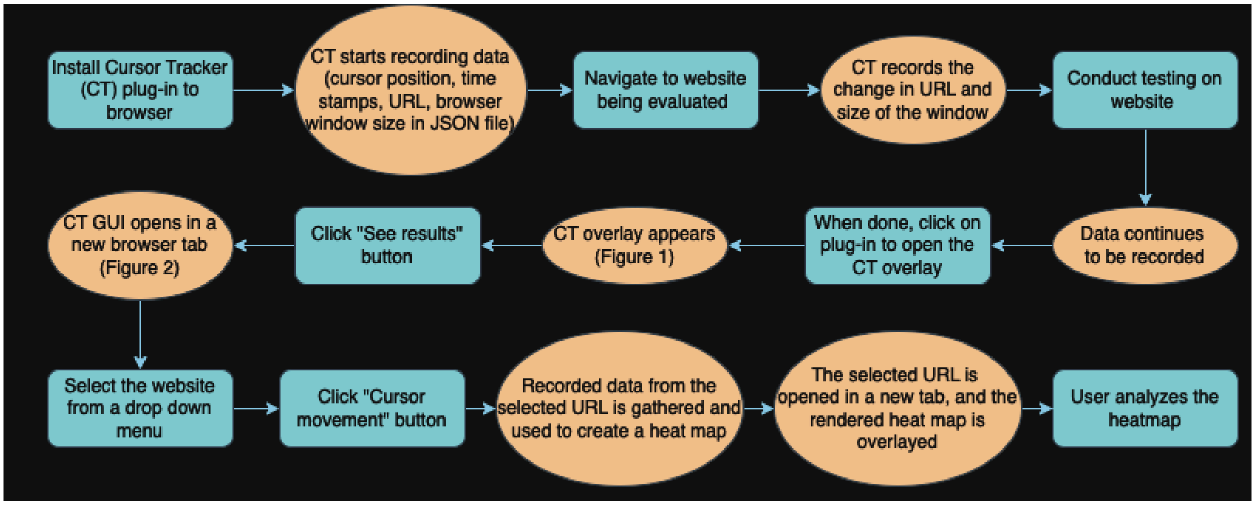

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Tue Sep 24 1:31 PM
Finexo
New Tab
File /Users/edwardcody/Downloads/finexo-html/index.html
Finexo
HOME
ABOUT
SERVICES
WHY US
TEAM
LOGIN
CRYPTO
CURRENCY
Read More

Explicabo esse amet tempora quibusdam laudantium, laborum eaque magnam fugiat hic? Esse dicta aliquid error repudiandae earum suscipit fugiat molestias, veniam, vel architecto veritatis delectus repellat modi impedit sequi.

Technical "flow"


User Input
Presses "Clicks" heat map button
Presses "Cursor movement" heat map button
URL selected in dropdown
Initiates
Initiates
Matches
Processes
Create Cursor Movement Heat map (using visual-heatmap)
Create Clicks Heat map (using visual-heatmap)
Rendered using
Rendered using
Data (saved to JSON file)
Cursor position (measured in pixels)
Clicks (boolean)
Time stamps (measured in milliseconds)
URL
Window size (measured in pixels)
Used if..
Creates
Creates
Output
Cursor Movement Heat map PNG
Create Clicks Heat map PNG

Example of an interaction (Blue is user action. Orange is happening behind the scenes)


Install Cursor Tracker (CT) plug-in to browser
CT starts recording data (cursor position, time stamps, URL, browser window size in JSON file)
Navigate to website being evaluated
CT records the change in URL and size of the window
Conduct testing on website
Data continues to be recorded
When done, click on plug-in to open the CT overlay
CT overlay appears (Figure 1)
Click "See results" button
CT GUI opens in a new browser tab (Figure 2)
Select the website from a drop down menu
Click "Cursor movement" button
Recorded data from the selected URL is gathered and used to create a heat map
The selected URL is opened in a new tab, and the rendered heat map is overlayed
User analyzes the heatmap

Final (self) assessment

- After working through the spec, what was the biggest most unexpected change to had to make from your sketch?
  - I realized that the size of the browser window will likely be important. I believe that visiting the same website two times, but with different-sized browser windows, will affect the heatmap overlay. For version 1, Cursor Tracker will just record the size of the browser window. If this assumption is right, later versions of Cursor Tracker will use the window size data to ensure the heat map overlay displays properly.
- How confident do you feel that you can implement the spec as it's written right now?
  - Coding has never been my strong suit, so I would guess that I have a 75% chance of being successful. I tried to keep this first version fairly simple, while also laying the groundwork for a more complex software in future versions.
- What is the biggest potential problem that you NEED to solve (or you'll fail)?
  - How to get the heatmap to render properly using visual-heatmap.
  - Notable mentions:
    - How to extract the size of the browser window.
    - How to save data locally from a browser plug-in.
- What parts are you least familiar with and might need my help?
  - This is a lot of uncharted territory for me. I can't really provide a specific area that I think I'll need help with yet. Once I get fully started, I will have a better idea.