# SPAR Eye-Tracking Pupil Labs UX Design Audit

## Targetability

Eye-Tracking with Pupil Labs requires a larger target UI size for accuracy than head pose. SPAR head pose used smaller UI target, determined based on legibility + targetability (a separate exercise would need to be taken to determine smallest usable head pose target size, but that is outside of the scope of this audit – we'll focus changes in UI scale based on increase from the current scale).

### Cursor Redesign

* Current circular cursor is not going to work for eye-tracking – needs further design exploration. We don’t want to show the cursor moving the whole time. Examples in eye-tracking we’ve seen that work better Pupil Labs highlight the background around a focal point.
* The impacts on dwell visualization are unclear without further exploration to see whether current racetrack is a distraction. Would need to be tested. Re-design may be needed.

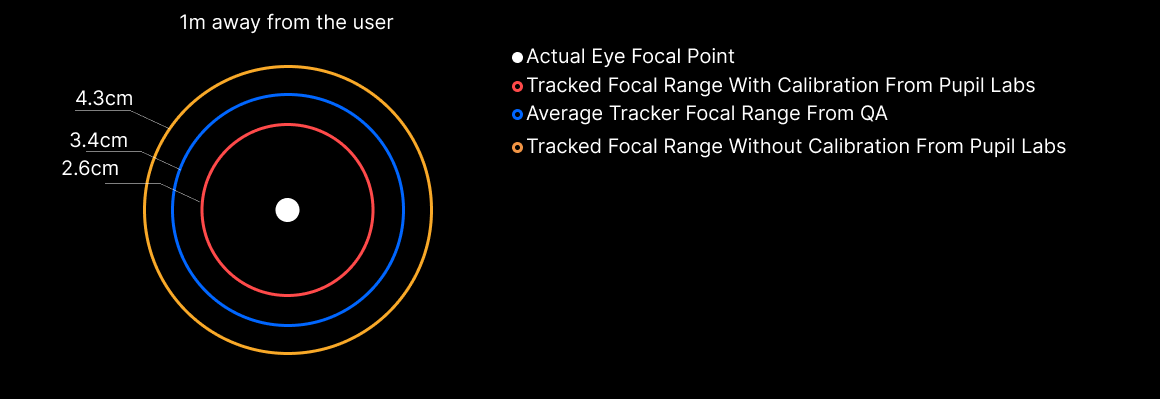
### Activation

* Selecting a button (click equivalent) will likely be based on a timed dwell, as with current head pose approach.
* With keyboard keys specifically we may also explore swipe with eye-gaze for selection of multiple characters.
* Blink to select is something we should rule out. It performed very poorly with Pupil Labs. Results in a very jump cursor experience, which we found used.

### Target Size

At a distance of 1M away from the AR content on our current display:

* Tracked focal range with calibration (according to Pupil Labs) is 2.6cm
  + 1.5 degrees radius from focal point
* Average tracked focal range with calibration (according to QA) is 3.4cm
  + 2 degrees radius from focal point
* Tracked focal range without calibration (according to Pupil Labs) is 4.3cm (WORST CASE SCENARIO -- WHAT WE ARE DESIGNING FOR)
  + 2.5 degree radius from focal point

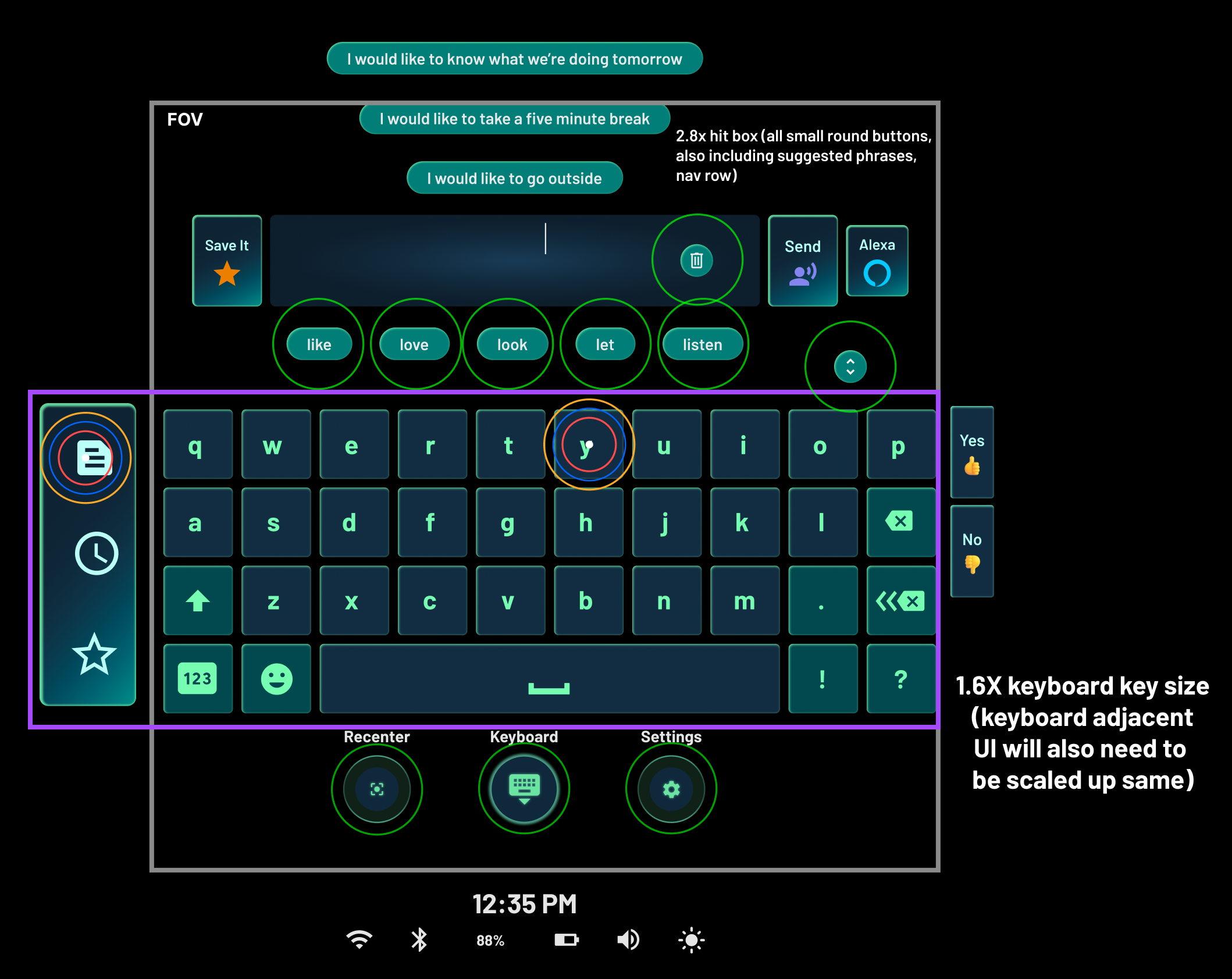


### UI Element Scale

To accommodate the above tracked focal range without calibration (worst case scenario) the following scale adjustment would need to be made to existing SPAR UI elements to be targetable.

#### QWERTY Keyboard Screen

* Keyboard keys and adjacent buttons to left and right (e.g. Phrase side nav bar and yes / no buttons) need to be increased 1.6x in size. **Thankfully this allows the core QWERTY keys to just barely fit in in within the width of the FOV**, but would cut out the UI on the side.
* Text field and surrounding UI there would need to be increased 1.6x in size.
  + Exception: clear button (or its hit target) would need to be increased 2.8x
* Suggested words (below) and phrases (above) would need to be increased 2.8x
* App Nav Bar UI (e.g. Re-center) can be left at the current scale 1x, though hit targets would need to increase to tracked focal range
* Status Bar can left at current scale of 1x, assuming it remains non-interactive



#### Phrase Manager Screen

* All buttons on panel would need to be scaled up by 2.8x
* All other buttons would be adjusted as described in QWERTY section

## 

## UI Element Position

Some of the current UI elements will no longer be able to fit within the FOV as a result of the UI scaling changes. Position of these would need to redetermined to keep them within FOV. It may also be difficult to target UI elements close to the edge of the FOV (need to test to determine), however this may not pose an issue in some cases if we have tagalong system, as opposed to fixing UI within the FOV.

#### QWERTY Keyboard Screen

* Phrase side nav bar
* Yes/No buttons
* Status Bar
* Phrase Suggestions

#### Phrase Manager Screen

* Up-scaling overall Panel by 2.8x will make it way too big – not a good approach
* Panel UI should fit in the FOV at current size, as long as UI elements inside are scaled is described in the previous section.

## Miscellaneous

* In general as a result of UI scaling and position changes, much of the UI will need to be re-spatialized and in some cases re-designed.
* We’ll need to update entire visual design system for eye-tracking UI.
* We’ll be able to re-use the headpose tagalong system, but dialed for eye-gaze (i.e. small movements for targeting, coarse movement for resetting overall UI context in FOV), so hopefully less-rework there, though at a minimum some tuning would be required.
  + Tagalong will also provides the benefit (unlike fixed position UI) of allowing a user to have a UI element show up in different parts of their FOV, as opposed to one location, in the event that looking in certain places with the FOV is difficult or uncomfortable.
* Need a persistent eye tracking status indicator and potentially notifications to let user / caretaker know when eye-tracking is not working.