Rubik’s Team Progress

12 May 2021

## Current Features Yet to Be Implemented

1. Data export
   1. Some of the data does get exported as expected, but the most critical ones are absent:
      1. Number of moves taken to solve the cube
      2. True time taken to solve the cube (Qualtrics’ timer records clicks on the page and time spent on the page, not related to the simulator itself)
      3. If the current step is solved or not
   2. As it stands right now, to get extra data to export, Embedded Data is the feature Qualtrics uses. The problem we faced, which ultimately rendered this project incomplete, is that the simulator survey’s embedded data does not transfer to other surveys, and for unknown reasons, we cannot add embedded data at runtime **and** still have it saved to export. Qualtrics documentation has been hit-or-miss (mostly miss) for us, so we could not find out why Qualtrics.SurveyEngine.addEmbeddedData(‘<nameOfData>’, ‘<valueOfData>’) would not export with the rest of the survey.
2. Timer
   1. The timer is present in Qualtrics, but it does not accurately record the time the participant takes to solve the cube. It does not save the time when the cube is first solved.
3. Sandbox mode
   1. Still checks the cubelets when the sandbox option is selected (as shown in the developer console).
   2. There should be a way to indicate in the exported data if the cube was in sandbox mode.

## Known Bugs/Design Flaws/Possible Issues

1. Shuffle curtain not synchronized to the shuffle sequence
2. To prevent the participants from seeing the cube shuffling, we put up a <div> element that blocks the shuffle sequence. After shuffleSequenceLength \* cubesAnimationSpeed milliseconds of setTimeout, the curtain is hidden. The bug is as follows:
   1. The setTimeout() can still be counting milliseconds in the background if the tab is not active, so the curtain can drop before the shuffle sequence is complete.
   2. Also, the curtain is based on a fixed position on the page, not relative the simulator container (the black box that holds the Rubik’s cube) itself. It should properly cover the cube.
3. A separate <div> tag is used to notify the participant that the cube is solved, but it does take up more space than needed. It adds whitespace between the submit button and the bottom of the simulator container when no longer hidden.
4. The keyboard commands feel backwards (Shift+<key> for clockwise rotations and <key> for counter-clockwise rotations) after switching back and forth between the button and keyboard controls. Check with Dr. Meinz on this.
5. There is no visual way to indicate if the simulator container will receive keyboard input (the participants must first click on the cube or the simulator container with their mouse before being able to use keyboard controls)