Solving Complex Problems Remotely: Does the Device Matter?

Timotej Iliev
Dr. Leventhal

ABSTRACT

In 2020, more exams were completed online than any other year before. As students sat behind a screen, we asked ourselves, does the device they use matter? Would their approach differ if they were to use a touchscreen, or perhaps a gamepad? Would we see an increase or decline in test scores, or would there be no difference at all? This research aims to answer all of those questions.

INTRODUCTION

Our objective is to create an interactive software package to administer and collect responses for Raven's progressive matrices – a nonverbal, standardized intelligence test. The software will remotely gather usage data, giving us insights into the problem-solving process and its link to the device being used.

Since the software had to be compatible across different platforms, the obvious approach was to create a web-based app. In order to do so, we used HTML, CSS and JavaScript, in combination with the JavaScript library interact.js

Interact.js allows the building of elements that are interactive, more specifically draggable. In addition, interact.js allows for interactivity on touchscreen devices as well as desktops.

ROAD BLOCKS

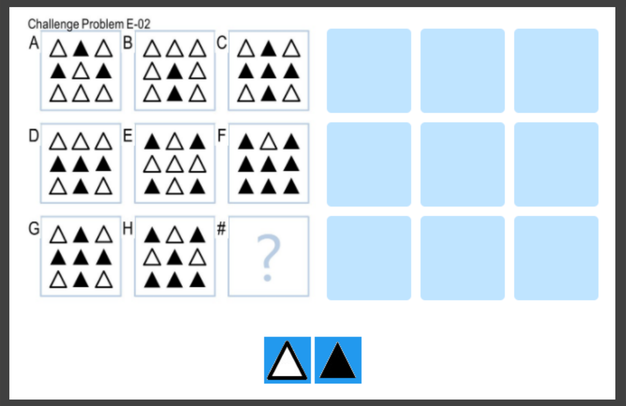
Throughout the implementation of this project there were many technical challenges.

Firstly, we explored different approaches which we could use to implement the software. The main dilemma was the usage of React Native. We decided that it might be best to approach it in a simpler matter, where we could just utilize an existing JavaScript library that would address most of our concerns.

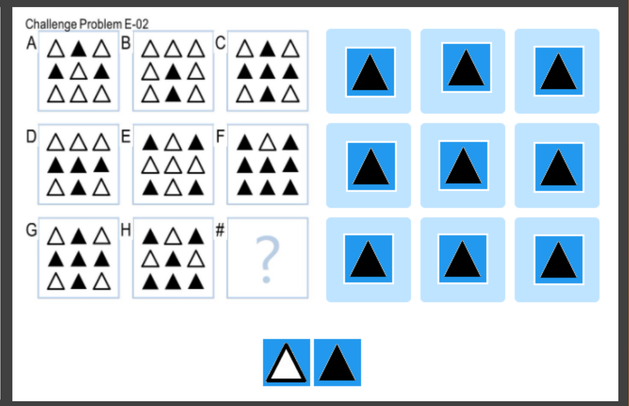
There were many libraries that seemed promising. The main contender was draggable JS. We found that it was difficult to use and that it did not have everything we would require for our purposes.

This is one of the main reasons we decided to explore and use interact.js.

INTERACTIVE RAVEN'S EXPERIENCE



The user is presented with a 3x3 matrix. Based on the information presented they have to determine the final part of the pattern and use the toolbox below to drag and drop the components to their corresponding placement.



The user drags and drops the component into the empty 3x3 matrix to solve the problem.

SOFTWARE

REQUIREMENTS

- Interactive Raven's task (web-based, cross-device compatible)
- Needs to support a variety of Raven's patterns
- Needs to remotely collect usage data
- Needs to able to detect as users take elements out of the toolbox, place it into the solution matrix and remove it from the solution matrix

SOLUTION

- Used interact.js, a JavaScript library that allows for cross-device compatibility
- Because the software was developed in JavaScript, it can be easily accessible over the web, enabling remote usage and data collection
- The element movement detection is done by using JavaScript listeners

University of Findlay

CONCLUSION AND FUTURE PLANS

- Although we were able to develop most of the software, we were not able to collect user data
- The next step would include:
  - Working on implementing additional raven's matrices
  - Working on the user experience, adjusting things such as auto-centering the elements dragged into the solution matrix.
  - Conducting the experiment and conclude if there would be a different solving approach based on the device used

REFERENCES

Bocanegra, B. R., Poletiek, F. H., Ftitache, B., & Clark, A. (2019). Intelligent problem-solvers externalize cognitive operations. *Nature Human Behaviour, 3*(2), 136-142. doi:10.1038/s41562-018-0509-y

Carpenter, P. A., Just, M. A., & Shell, P. (1990). What One Intelligence Test Measures: A Theoretical Account of the Processing in the Raven Progressive Matrices Test. *Psychological Review, 97*(3), 404-431. doi:10.1037/0033-295x.97.3.404

Lovett, A., & Forbus, K. (2017). Modeling visual problem solving as analogical reasoning. *Psychological Review, 124*(1), 60-90. doi:10.1037/rev0000039

Matzen, L. E., Benz, Z. O., Dixon, K. R., Posey, J., Kroger, J. K., & Speed, A. E. (2010). Recreating Raven's: Software for Systematically Generating Large Numbers of Raven-Like Matrix Problems With Normed Properties. *Behavior Research Methods, 42*(2), 525-541. doi:10.3758/brm.42.2.525