Scalemate: Psychological Tests Engine

Cristiano Silva Júnior

Abstract

Let's talk about this psychological tests engine, written by me. I wrote it using C#, trying to show that every test can be implemented using a queue and a simple sum system. Of course, that is not the real abstract.

Should there be keywords? Like, queue, free software?

1   Introduction

The Scalemate engine was conceived based on the idea that most psychological tests can be written in the form of a queue, as understood by computer scientists. That is, each item of that queue can represent a group of question and answers. As each item is answered, the queue can move and the next items appear to the subject until there are no more questions left.

Not every psychological test can be done this way: many of them follow complex algorithms that can not be simply implemented with a simple queue, but for those that can be done this way (notably, those that follow the Likert scale model), the Scalemate engine is available for use. It was designed for a simple user interaction, so that everyone, even those that are not used to computers, can make use of it and expand it with their own tests.

Most of the available tools for implementing psychological tests on the market require the researcher to know at least some concepts of computer science and algorithmics, and our experience with these tools show that, for most intended work, it is not necessary to know any of these concepts, so using this kind of software can be overwhelming to those not used to the thrills of computer programming. To fill this gap, we developed Scalemate to be a simple and smooth experience to researchers so they can proceed with their job with minimum stress.

Scalemate aims to help those that perform simple question-and-answer based research on places where internet is not readily available, thus making this platform completely offline; and where power supply may be a problem, thus making an lightweight application, consuming as much as 20MB of RAM memory, depending on how big your tests are.

2   Methods

Developed for Microsoft Windows using the C# programming language and available for free under the MIT license, Scalemate can be downloaded from GitHub, and everyone is invited to contribute to it. It was written with the Windows Forms technology from Microsoft Visual Studio.

The download consists of a compressed folder which, once extracted, contains an executable, made to run on computers running Microsoft Windows Vista onwards, and two folders: *assets* and *results*.

The *assets* folder is the place to store the tests per se. There is a file called "kinds.txt" to list every test that must be shown in the program. Each line of this file must follow the following organization:

- The first word is the the name of the folder containing a test.

- The following words are the test label, and will appear on the main screen for easier selection.

For each line in the "kinds.txt" file, there must a corresponding folder, containing the needed information for the test.

For each test, there must be at least two files inside its folder:

- A "inventory.txt" file containing the questions and answers for the test. The first line in this file must be the number N of possible answers in the test. The following lines must be organized in groups of N+1 lines: the first one is the question that will appear on the top of the test screen, while the following N lines are the possible answers. They must be placed according to their respective scoring: the first answer is worth 0 points, the second answer, 1 point; the third answer, 2 points; and so forth. If the researcher intends to revert this scoring logic, they can add a '*' character before the question sentence. The application will understand this as a command and this character will not appear in the questionnaire but will influence its final result.

- A "results.txt" file which will describe how to score this test. Each line must contain three itens: the first one is a number, representing the lowerbound of this specific result; the second item is another number, defining the upperbound; and the rest of the line is the result to be written in the test result. These must separated by space characters. Scalemate will score a test application by summing the subject's points and checking if their score is equal or greater than the lowerbound and less than the upperbound. When that is the case, the engine will consider that as the correct result. When designing a test, the researcher must be sure that there is no overlapping between each result boundary and that every possible score is described in this file.

There are two other files that can appear inside the test's folder, but they are optional. They serve to make the test's application easier. These files are:

- A "information.txt" file with a list of questions to be asked for before the test begins, so the researcher can survey some other information about the subject before the assessment actually starts.

- A "instructions.txt" file with a text to appear before the test's beginning to instruct the subject on how to proceed or to give whatever information the researcher considers important or useful for that procedure. If omitted, Scalemate will go straight to the test's first question.

The *results* folder on Scalemate is used to store the users' results, as saved and processed by the engine. After each test completion, the application will generate a tab-separated values file, unbundling the subject's information, their result on the test and each given answer. Each file is named as {*subject name*}_{*test folder*}.*csv* and is organized as follows:

- The first line contains the information needed to identify the subject. It reveals the test's folder name, the subject name, their score and the result given by Scalemate, in this order.

- If the "information.txt" file is present, then the following line contain the answers to the initial survey.

- The following line has the answers to each question on the test, for later analysis. Each answer represents the index minus one of the selected item, that is, if the user chose the first option, then the number 0 will be stored to represent that answer; if they chose the second option, then the number 1 will appear, and so forth.

3 Results

In order to demonstrate Scalemate's data collection abilities, we conducted a small research based on a paper by Clarry Lay [1] on procastination. We applied his questionnaire using this program and developed a small MATLAB application to analyze the generated data.

We applied the test to $N = $ inf students from the University of Brasilia from many differents graduation programs and backgrounds to find a correlation between their average academic performance and their scores on Lay's scale.

Bushwick slow-carb authentic wolf. Crucifix gentrify put a bird on it kinfolk, disrupt vice artisan pinterest distillery mlkshk affogato raw denim portland seitan. Blue bottle humblebrag bitters, ugh lomo pinterest artisan pork belly post-ironic dreamcatcher. Kitsch kale chips wayfarers, chicharrones taxidermy polaroid roof party YOLO schlitz slow-carb occupy literally. Ugh irony gluten-free, scenester trust fund tousled cornhole celiac keytar dreamcatcher mumblecore ethical. Literally 90's lomo fingerstache. Taxidermy vegan selfies, hella offal portland pug biodiesel gluten-free synth four dollar toast echo park tousled.

Plaid helvetica heirloom, letterpress mixtape kale chips keffiyeh sriracha affogato fap beard mumblecore. Messenger bag tumblr put a bird on it flannel neutra, irony iPhone forage lo-fi pop-up. Occupy pop-up tousled chartreuse, fanny pack VHS plaid blog. Pinterest health goth kitsch tousled etsy williamsburg art party offal affogato cliche bushwick poutine small batch. Ramps biodiesel dreamcatcher art party, cold-pressed ennui taxidermy shoreditch chambray. Twee venmo celiac austin, 8-bit trust fund tousled taxidermy pitchfork. Shabby chic green juice fingerstache forage tofu.

Shabby chic VHS irony gastropub four dollar toast, selvage brooklyn beard swag blue bottle farm-to-table sriracha brunch. Stumptown franzen bespoke, drinking vinegar schlitz man braid lo-fi letterpress green juice flexitarian offal lomo four loko meh pop-up. Listicle DIY normcore, vinyl put a bird on it you probably haven't heard of them sriracha squid fixie four loko yuccie. Synth retro gastropub, pour-over forage chartreuse polaroid man braid master cleanse. Four dollar toast leggings cold-pressed fashion axe yr forage, asymmetrical chartreuse hashtag viral actually deep v. Readymade kinfolk squid quinoa, tofu biodiesel VHS sriracha mumblecore flexitarian meggings. Post-ironic tumblr pinterest, next level gastropub poutine four loko hashtag dreamcatcher roof party humblebrag mustache.

Mumblecore bitters austin dreamcatcher man bun ugh. Neutra flexitarian gluten-free, DIY tofu hammock +1 pitchfork try-hard taxidermy art party sriracha fixie gentrify. Hoodie lumbersexual tousled asymmetrical. Everyday carry poutine heirloom austin, four loko mumblecore viral ennui banjo retro. Franzen green juice chillwave offal bitters fap, keytar mixtape. Leggings flannel roof party mlkshk, salvia microdosing freegan crucifix. Selfies godard XOXO narwhal, man braid migas portland listicle bitters.

4 Future Works

Write about adding a test editor. And images.

Just as it is available, Scalemate can only be used for simple tests, and still require the research to open and deal with many files. Therefore the most obvious steps are:

- Enable multimedia content inside the tests, like images and sounds. This will expand the number of fields in which Scalemate can be used;

- Create a user-friendly test editor whithin the program, so the researcher does not need to leave the application to deal with test data or with gigantic text files.

Another possible improvement can be done on the way Scalemate process the collected data. Right now, it just uses a simple score based grade system, whereas some other more complex analysis could be done on the collected information. Even though Scalemate's scope covers mostly how the researcher will apply their tests, it can also come really handy to have an inside tool to help with further data mining and processing.

5 Bibliography

References

[1] Clarry Lay, At Last my research article on procrastination, Journal of research in personality 20, 1986.