The Impact of Visual Load on Performance in a Human-computation Game

Christina Chung[1], Amit Kadan[1], Yueti Yang[1], Asako Matsuoka[1], Julia Rubin[2], Marsha Chechik[1]
University of Toronto, Canada[1]
University of British Columbia, Canada[2]
{chr.chung,amit.kadan,elsie.yang,asako.matsuoka}@mail.utoronto.ca,julia.rubin@ece.ubc.ca,chechik@cs.toronto.edu

ABSTRACT

It is well-known that tasks imposing high cognitive load, i.e., the mental effort required to carry out a task, place a strain on people's ability to perform. In light of this, the present study investigates whether poor performance also occurs in human-computation games. That is, do players perform better in game designs that increase the visual information presented? These designs have the advantage of exposing players to more of the solution space, but may come with the caveat of imposing a higher cognitive load. We present a case study by considering alternative layouts differing in the amount of visual information given to players in a human-computation game. The findings of the study seem to support the idea that presenting more information is beneficial to players. This is surprising result that challenges prevailing beliefs about cognitive load, and invites more detailed, future investigation.

CCS CONCEPTS

•**Human-centered computing → Empirical studies in visualization;** *Empirical studies in collaborative and social computing;* Visualization design and evaluation methods;

KEYWORDS

Human-computation games, human computation, serious games, games with a purpose, game design, information visualization

ACM Reference format:
Christina Chung[1], Amit Kadan[1], Yueti Yang[1], Asako Matsuoka[1], Julia Rubin[2], Marsha Chechik[1]. 2017. The Impact of Visual Load on Performance in a Human-computation Game. In *Proceedings of FDG'17, Hyannis, MA, USA, August 14-17, 2017,* 4 pages.
DOI: 10.1145/3102071.3106358

1 INTRODUCTION

Human-computation games have emerged as a promising way of employing human abilities to solve problems that are difficult for computers. While traditionally used to solve cognitive problems such as image tagging [18], in recent years, games have shown success in solving computationally expensive problems, which include protein folding [6], protein sequence alignment [11], software verification [8], and model merging [5].

FDG'17, Hyannis, MA, USA

DOI: 10.1145/3102071.3106358

To solve these problems, the games typically leverage human pattern recognition abilities by encoding the problems in a visual manner. As an example, a game for protein folding known as *Foldit* presents proteins in their three-dimensional molecular form, for which players must use their visual abilities to identify optimal configurations [6]. Analogously, the protein alignment game described by Kawrykow et al. requires players to match colorful blocks which encode protein sequences [11]. Dietl et al. developed a game for software verification that represents programs by a network of pipes and balls. Players derive a proof of correctness by aligning the pipes to allow all of the balls to freely travel through them [8].

In light of the increasing importance of human-computation games for solving complex problems and their heavy reliance on human visual abilities, it is imperative to understand how player performance interplays with the amount of visual information presented.

It is well-known that human working memory has a finite capacity; storing only a few elements of information at any given time [12], and if not retained, the information is lost within thirty seconds [7]. In addition to storing information, working memory is responsible for processing information during cognitive tasks. A direct implication to this is that if many elements are stored in working memory and if the task at hand is demanding, then cognitive overload may occur (whereby the mental effort required to complete a task exceeds the brain's capacity), resulting in poor performance.

This corroborates with empirical evidence showing impediments on performance during tasks which impose high cognitive load [2–4, 15]. This may suggest that game designers should reduce the visual information given to players to preclude the possibility of hindering performance.

Yet, inducing higher cognitive load has no benefits for the tasks described in these studies. For instance, employing ineffective problem-solving strategies [15] and instruction manuals [4] does not mediate performance in any way. In contrast, players may benefit from seeing more information in a human-computation game. Seeing more exposes players to more of the solution space, allowing them to identify better solutions. In the case of protein folding, being able to see the entire protein helps players identify the best places at which to configure them. As described in the literature, players still perform well even for proteins that are large and complex [6].

Presently, it is unclear as to whether designers should attempt to minimize or maximize the visual information given to players. Would players benefit from seeing more information, or would the higher cognitive load place too much of a burden on their ability to perform? As noted in the literature, research in human factors for

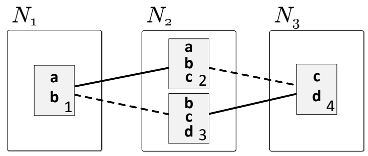

Figure 1: An example of n-way matching on inputs N_1, N_2, and N_3.

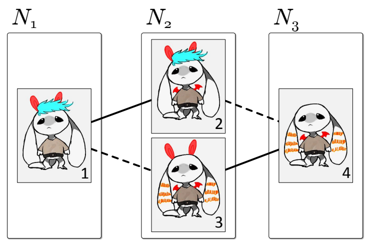

Figure 2: The representation of elements from Figure 1 in MATCHMAKERS.

guiding these designs is wanting [9, 10, 16]. Given that these games rely on human visual abilities, knowing how much information should be presented in order to optimize player performance is of paramount importance to designing a successful game.

We conduct a case study to address this question using MATCHMAKERS [5], a game that solves a computationally-expensive problem known as n-way matching [14].

We begin by describing MATCHMAKERS and n-way matching in Section 2. We then outline the research question, the design of the study and disseminate the results in Section 4. In Section 5 and 6, we discuss the study's implications and limitations. Finally, we conclude in Section 7.

2 MATCHMAKERS

MATCHMAKERS [5] is a game that aims to solve n-way matching [14], the process of identifying similar elements between distinct inputs. N-way matching has applications in many software engineering practices, such as merging branches of a software configuration management system. Yet, the problem is NP-hard, and existing algorithmic approaches used to solve it either do not scale or do not guarantee high-quality solutions [14].

Figure 1 depicts an example of matching the elements in inputs N_1, N_2, and N_3. N_1 has one element with properties *a* and *b*; N_2 has two elements, one with properties *a*, *b*, and *c* and one with properties *b*, *c*, and *d*; N_3 has a single element with properties *c* and *d*. Note that n-way matching consists of binding elements from distinct inputs and therefore elements from the same input cannot be bound together. For example, it would be invalid to bind element *2* with element *3* in Figure 1.

Two different approaches to matching the elements are indicated by dotted and solid lines. One approach, represented by the dotted lines, binds elements *1* with *3*, and elements *2* with *4*. In that approach, the elements of each created pair share one property out of four. That is, elements *1* and *3* share property *b*, but not *a*, *c*, and *d*. A better approach is represented by the solid lines, which bind elements *1* with *2*, and elements *3* with *4*. In that approach, the elements of each created pair share two out of three properties; namely, elements *1* and *2* share properties *a* and *b*, but not *c*. This approach is, in fact, an optimal solution for the n-way matching problem, whose goal is to bind elements that share the largest number of common properties.

The game uses human visual cognition to perform visual comparisons quickly and efficiently. Input elements are encoded as an alien. For example, Figure 2 depicts aliens that represent the elements in Figure 1. These aliens are composed of items that represent the properties of their corresponding elements. The leftmost alien in Figure 2 encodes element *1* in Figure 1 by representing property *a* with a blue wig and property *b* with bunny ears. Resultingly, input elements sharing common properties are represented by visually similar aliens, and elements that have different properties are represented by visually different ones.

The game is played iteratively. It starts from the best known matching solution produced so far, either by a state-of-the-art heuristic algorithm or by a previous player. A player is presented with a target alien for whom they must "find friends". Players may scroll through the rest of the aliens and select aliens one by one to add to their created group. If the player manages to form a group of aliens that is deemed to be an improvement to the matching solution, the player wins and that solution is stored for the next iteration of the game. The player then restarts the game with a new target alien.

3 RESEARCH QUESTION

Working memory is severely limited in the amount of information it can hold and in its duration [7, 12]. A direct consequence to this is that for tasks which are cognitively demanding, we can expect memory overload to occur and performance to deteriorate.

We can therefore reasonably assume that players will not perform at their best when bombarded with visual information. In spite of this, an opposing hypothesis can also be drawn. Seeing more visual information exposes players to more of the solution space, which increases their chances of identifying optimal solutions. With respect to n-way matching, seeing more elements enables players to identify better groups of friends. This is evidenced by a number of games, in which players still outperform state-of-the-art approaches despite being presented with more information [6, 8, 11].

The following question now remains: *do players perform better when given a higher or lower visual load?*

4 EXPERIMENT

4.1 Methods

We adapt the work of the authors of [5] to investigate the question of whether players perform better with a higher or lower visual load. Our study considers two alternative game layouts for MATCHMAKERS: one which aims to increase the amount of information presented and one which aims to reduce it. Both layouts are available online [1].

Design *L1*, shown in Figure 3 (a), maximizes the number of aliens that can be shown to players. At any given time, about thirty aliens are presented on the screen (attempting to fit any more would make the aliens too small). The target alien for whom the player must find friends, is shown at the top of the screen. The aliens encoding

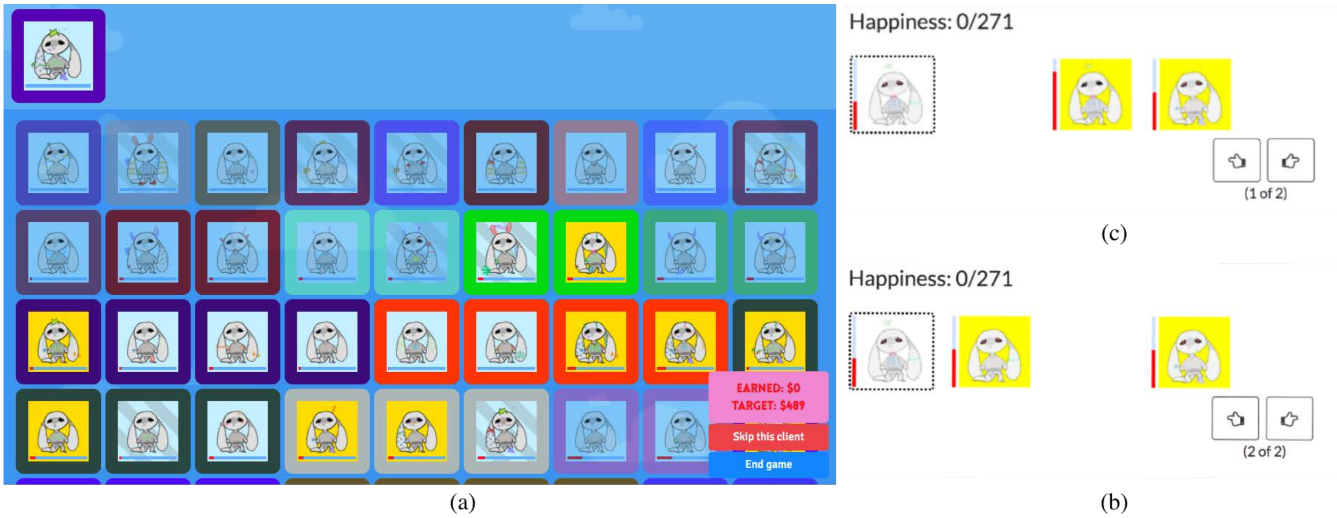

Figure 3: (a) A snapshot of layout *L1*; (b) a snapshot of *L2*; (c) a snapshot of *L2* after pressing the right arrow button.

the rest of the input elements are laid out below it on a grid. Aliens of the same group are juxtaposed. (As described in Section 2, aliens have already been placed into groups prior to each game based on the best solution produced by a previous player or a heuristic algorithm.) Group membership is indicated by the alien's border color, i.e., the two aliens on the lower right both have pink borders and therefore come from the same group. Players may scan through the aliens using the scroll bar on their web browser window, and select aliens to add to the new group of friends that they are forming by clicking on them.

The alternative design *L2*, shown in Figure 3 (b) and Figure 3 (c), reduces the information presented. The aliens are laid out along a single row. Similar to *L1*, aliens are juxtaposed with their group members. But rather than displaying many groups simultaneously, *L2* only displays a single group at a time for the purpose of reducing visual load.

Each index along the row displays aliens from a single input. The order of the inputs are randomized (with the exception of the target alien's input, which is always assigned to index 0). For example, the snapshot of *L2* in Figure 3 (b) depicts a case with five inputs used for the game's tutorial. The row contains five indices, one for each input. The target alien is at index 0. Index 1 displays the aliens from the second input; index 2 displays the aliens from the third input, and so on. On the screen, a group with two aliens is shown to the player, one at index 2 and one at index 3. Since the group does not contain aliens from the second and fifth inputs, index 1 and index 4 are left blank. Players may browse through all of the groups using the left and right arrow buttons shown on the screen. Figure 3 (c) shows the new group that is displayed when clicking the right arrow button from the game screen in Figure 3 (b). This group contains two aliens, one from the second input in index 1 and one from the fourth input in index 3. The group does not contain aliens from the third and fifth input, so index 2 and 4 are left blank.

In both designs, aliens that are deemed similar to the aliens in the group created by the player are highlighted with a yellow background using a heuristic algorithm. This significantly reduces the size of the search space, in that players only need to consider aliens that are highlighted. As another player aid, the red bar placed below each alien in *L1* and to the left of each alien in *L2* indicates the strength of similarity of the alien with other members of its group. A full bar means that an alien is very similar, while an empty bar means that it shares few characteristics with its group members.

4.2 Procedure

Participants were recruited through Facebook and Reddit. The highest-scoring players were rewarded with $100 in gift cards.

To evaluate the two layouts, we launched the game on a real-world case consisting of eight inputs that encode a system of hospitals reported in [13]. In total, there were 221 elements and 158 properties. Participants were initially randomly assigned to one layout, but were later permitted to play both. In total, thirty-seven participants took part in the study. Eleven players attempted *L1* and twenty-six players attempted *L2*. Fewer players attempted *L1* due to a bug that failed to load the game for some of the participants. The game was launched for a period of ten hours.

4.3 Results

We used two metrics to measure the layouts against each other. First, we considered the cumulative score of players, i.e., the improvement in score made by players to the n-way matching solution (*M1*). The score was measured using the weight function defined in [14]. We also measured the number of successful games normalized by the number of players, i.e., the total number of wins divided by the number of players (*M2*). The results are shown in Figure 4 and Figure 5.

Players did better on L1 based on both of our metrics. *M1* was 886 points for *L1* and 490 points for *L2*, meaning that the players of the first design were able to improve the score of the n-way matching by 886 points – more than the players of *L2* even though there were many more of them! In addition, *M2* was 0.36 for *L1* and 0.31 for *L2* indicating that there were more individual wins for the first design than the second. Putting the two together, we can conclude that in

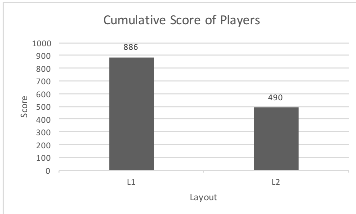

Figure 4: The cumulative score achieved by players.

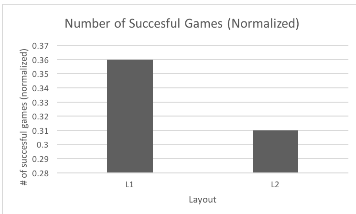

Figure 5: The number of successful games normalized by the number of players.

our experiment, participants who had more information were more likely to score a win than those who received less information.

The results of the study suggest that players benefit more from a higher visual load. Players not only achieved a higher cumulative score in *L1* (*M1*), but they were also more successful in their attempt to improve the n-way match solution (*M2*). *L1*, which displays many aliens on the screen and makes it easy to quickly scan through them all, enables players to look for the best out of all possible aliens to add to the created group. On the other hand, *L2* presents only a small number of aliens at a time, and thus cannot facilitate such a strategy. Instead, players likely employ an approach in which they will add the first alien that makes some (but not necessarily the most) improvement to their group.

5 THREATS AND LIMITATIONS

The findings of the study should be taken with some precaution. Firstly, it did not consider the performance of individual players, whether or not the order in which they played the two layouts had any bearing on performance, nor the amount of time they spent playing. The results are also threatened due in part by the study's short duration and the small number of players who participated.

While the results suggest that *L1* is better than *L2* for helping players achieve success, there could still be some benefits to *L2*. For instance, if *L2* is significantly less overwhelming than *L1*, players might enjoy playing a game employing *L2* more than one employing *L1*. This could prove to be significant in the long run, as it is crucial to retain an active player base for a game to garner success [17]. *L2* can also reach a wider array of players, since presenting a small number of elements in *L2* makes it possible for the game to be played on mobile devices. In contrast, *L1* can only be played on a large display.

In addition, we did not consider the impact of player aids on performance. As mentioned in Section 4.1, the bars and highlighting act as visual cues that substantially reduce the size of the search space. It would be interesting to explore whether the findings will be similar in the absence of aids.

6 CONCLUSION

Due to the limited capacity and duration of human working memory, tasks demanding high cognitive load tend to cause poor performance. However, it is not clear whether this holds in human-computation games, where seeing more visual information can be advantageous by exposing players to more of the solution space. To our surprise, the present study seems to support the notion that more information is of greater benefit. Still, the results must be interpreted with some caution due its limitations; further investigation using more players, other problems and a longer duration is needed to arrive at a more definitive conclusion.

7 ACKNOWLEDGEMENTS

We thank Angel You and Si Hua Cao Liu for their contributions to the development of earlier versions of the game, and Fei Huang for her drawings of the game characters. We also thank anonymous reviewers for their valuable advice.

REFERENCES

[1] 2017. MATCHMAKERS. https://anonymoususer12.github.io/. (May 2017).
[2] Paul Chandler and John Sweller. 1991. Cognitive load theory and the format of instruction. *Cognition and instruction* 8, 4 (1991), 293–332.
[3] Paul Chandler and John Sweller. 1992. The split-attention effect as a factor in the design of instruction. *British Journal of Educational Psychology* 62, 2 (1992), 233–246.
[4] Paul Chandler and John Sweller. 1996. Cognitive load while learning to use a computer program. *Applied cognitive psychology* 10, 2 (1996), 151–170.
[5] Christina Chung, Asako Matsuoka, Yueti Yang, Julia Rubin, and Marsha Chechik. 2016. Serious Games for NP-hard Problems: Challenges and Insights. In *Proc. of the ICSE Workshop on Games and Software Engineering (GAS@ICSE'16)*. 29–32.
[6] Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović, and others. 2010. Predicting protein structures with a multiplayer online game. *Nature* 466, 7307 (2010), 756–760.
[7] N. Cowan. 2001. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioural Brain Sciences* 24 (2001), 84–114.
[8] Werner Dietl, Stephanie Dietzel, Michael D. Ernst, Nathaniel Mote, Brian Walker, Seth Cooper, Timothy Pavlik, and Zoran Popovic. 2012. Verification Games: Making Verification Fun. In *Proc. of the ECOOP Workshop on Formal Techniques for Java-like Programs (FTfJP'12)*.
[9] Ivan Herman, Guy Melançon, and M Scott Marshall. 2000. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on visualization and computer graphics* 6, 1 (2000), 24–43.
[10] Weidong Huang, Peter Eades, and Seok-Hee Hong. 2009. Measuring effectiveness of graph visualizations: A cognitive load perspective. *Information Visualization* 8, 3 (2009), 139–152.
[11] Alexander Kawrykow, Gary Roumanis, Alfred Kam, Daniel Kwak, Clarence Leung, Chu Wu, Eleyine Zarour, Luis Sarmenta, Mathieu Blanchette, Jérôme Waldispühl, and others. 2012. Phylo: a Citizen Science Approach for Improving Multiple Sequence Alignment. *PloS ONE* 7, 3 (2012).
[12] George A Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review* 63, 2 (1956), 81.
[13] Y. T. Rad and R. Jabbari. 2012. Use of Global Consistency Checking for Exploring and Refining Relationships between Distributed Models: A Case Study. *Master's thesis* (2012).
[14] Julia Rubin and Marsha Chechik. 2013. N-Way Model Merging. In *Proc. ESEC/FSE*. 301–311.
[15] John Sweller. 1988. Cognitive load during problem solving: Effects on learning. *Cognitive science* 12, 2 (1988), 257–285.
[16] Melanie Tory and Torsten Moller. 2004. Human factors in visualization research. *IEEE transactions on visualization and computer graphics* 10, 1 (2004), 72–84.
[17] Luis von Ahn and Laura Dabbish. 2008. Designing Games with a Purpose. *Commun. ACM* 51, 8 (Aug. 2008), 58–67. DOI: http://dx.doi.org/10.1145/1378704.1378719
[18] Luis von Ahn, Ruoran Liu, and Manuel Blum. 2006. Peekaboom: A Game for Locating Objects in Images. In *Proc. of the Conference on Human Factors in Computing Systems (CHI'06)*.