Using Q-Learning for Sequencing Level Difficulties in a Citizen Science Matching Game

Sofia Eleni Spatharioti
Northeastern University
Boston, MA, USA
spatharioti.s@husky.neu.edu

Sara Wylie
Northeastern University
Boston, MA, USA
s.wylie@northeastern.edu

Seth Cooper
Northeastern University
Boston, MA, USA
scooper@ccs.neu.edu

CHI PLAY EA '19, October 22-25, 2019, Barcelona, Spain.

ACM ISBN 978-1-4503-6871-1/19/10.
https://doi.org/10.1145/3341215.3356299

Abstract
Sequencing task difficulty and variety can be a powerful tool for increasing engagement in online citizen science platforms. The abundance of available participant data presents great promise for machine learning oriented approaches to making tasks more engaging for participants. We present a web game for image matching called Tile-o-Scope Grid, and explore using a Q-learning based algorithm to generate a policy for sequencing level difficulties. Recruiting players using Amazon Mechanical Turk, we gathered data to train and evaluate approaches to sequencing level difficulties in Tile-o-Scope Grid. Comparisons of our Q-learning based algorithm with uniform random and greedy baselines suggest potential for using reinforcement learning for citizen science image labeling.

CCS Concepts
•Human-centered computing → Human computer interaction (HCI);

Author Keywords
citizen science; image labeling; Q-learning

Introduction
While a plethora of crowdsourcing platforms for citizen science image labeling have been proposed over the past two decades, ranging from disaster response [17, 1, 13, 12,

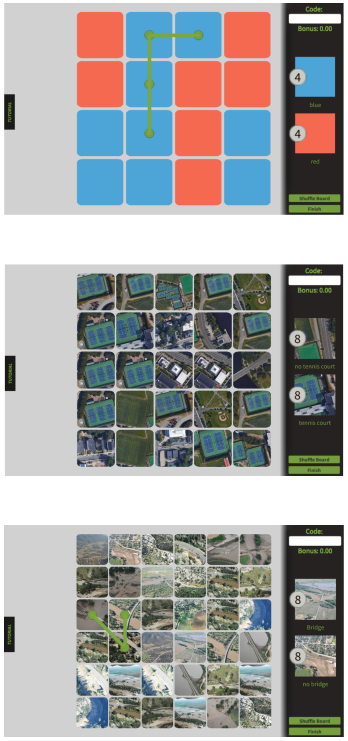

Figure 1: Example of the game interface, showing, from top to bottom, the difficulties *E* (easy), *M* (medium; Imagery ©2019 Google, Map data ©2019 Google) and *H* (hard; Images from U.S. Geological Survey).

24] to animal conservation efforts [19] and astronomy [15], dealing with participant disengagement, where participants quit contributing early in the process, remains a challenge [18]. As the classification of data requires a large and diverse pool of participants, designing effective and engaging interfaces is key to retaining participants for longer periods of time. The recent success of designing games for citizen science, along with the introduction of task variety, offers a promising avenue for designs that leverage gaming elements to maintain high participant engagement [4, 16].

However, many citizen science platforms offer limited to no elements of task variety, which have been shown to positively impact worker performance [8, 6]. When task variety is present, in the form of interleaving differing tasks in type or context, this is often achieved using a fixed sequencing scheme, ignoring participant performance. Recent work in using reinforcement learning (RL), a type of machine learning that uses training data to generate optimal policies for selecting actions, in level ordering in games, revealed great promise for the use of RL for sequence ordering design [11, 10].

We were interested in exploring RL algorithms to serve task difficulty sequences for players, i.e. the order in which a player encounters task difficulties, using game levels on a difficulty scale from easy to medium to hard. To this end, we developed an image matching game called Tile-o-Scope Grid, where images are placed on a grid and, similar to the game Dots [14], players are tasked with collecting images by drawing lines to connect images of the same category. We identified three levels of significantly varying difficulty, using three different datasets. We designed a difficulty sequencing approach based on the reinforcement learning Q-learning algorithm, to generate sequences of level difficulties for players. We then compared our Q-learning approach against both uniform random and greedy sequencing methods. Players were recruited by running Human Intelligence Tasks (HITs) on Amazon Mechanical Turk, a popular crowdsourcing marketplace, often used for recruiting participants.

We found that using Q-learning to sequence varying difficulties in Tile-o-Scope Grid may, in some cases, lead to not only higher levels of engagement, but also higher contribution of meaningful labels. Tile-o-Scope Grid is able to combine multiple datasets and generate difficulty sequences based on existing user data, instead of relying on static sequences. Our findings offer encouraging insights for using reinforcement learning to not only design more engaging game level orderings, but also to increase participation and performance in citizen science image labeling tasks.

Related Work

Collaborative image labeling games have long been used for achieving high quality labels and identifying objects in images [22, 23]. Notable examples of citizen science image labeling interfaces include Cropland Capture [18] and Snapshot Safari [19]. A tile-based game closely related to Tile-o-Scope Grid is Befaced [20], which aims to create a crowdsourced facial expression database. Players are tasked with making facial expressions that match the aligned tiles in order to successfully collect them. BeFaced deploys a Dynamic Difficulty Adjustment [7] algorithm to lower certain matching difficulties caused by certain facial expressions, as a means of retaining player engagement.

Utilizing task variety to increase engagement is the focus of a growing body of work. The impact of order in sequencing microtasks has been explored by Cai et al. [2], however, the chosen domain was not in citizen science and the microtasks did not involve image labeling. Research on con-

textual interruptions suggests that switching between tasks of different types may impact completion time [9]. The effects of task variety in a citizen science setting, specifically in disaster response, have been explored by Spatharioti et al. [16], where switching between tasks at specific intervals would lead to increased engagement, when measured as voluntary time. However, none of the above examined a reinforcement learning approach for generating automated sequences of task difficulty.

The high volume of data available makes games highly suitable for deploying RL algorithms. Mandel et al. explore various comparisons of RL algorithms using engagement in an educational game for evaluation of policy performance [11]. RL approaches were further deployed to combat player disengagement in Refraction, an educational game about fractions [10]. Q-learning algorithms were utilized in Q-DeckRec, a recommendation system for Collectible Card Games (CCGs) [3]. Q-DeckRec can be used in CCGs such as Hearthstone to suggest optimal deck builds that may lead, among other things, to increased player engagement. As the state space in CCGs is often too big for maintaining the lookup table required for Q-learning, a Multi-Layer Perceptron (MLP) approach is employed.

Study Setup

Game Setup

We developed a web game, called Tile-o-Scope Grid, which is similar to Dots [14], using Unity. In Tile-o-Scope Grid, tiles of images are placed on a 2D grid. The purpose of the game is to connect tiles that contain images of the same category by dragging a non-intersecting line connecting neighboring images in order to collect them. Diagonal lines are also allowed. Every level requires a specific amount of tiles to be collected of each category. If a line is valid and matches images of the same category (which can be checked for some images, which have ground truth categories associated with them), then the move is considered correct and the amount to be collected of the category is reduced by the number of tiles in the match. If the line contains images that do not belong to a unique category, then the move is incorrect and players are penalized, by adding items to their collection counts. Once the player has collected (at least) the required amount for all categories, the level is complete and the player can continue to the next. Players can also shuffle the board, in the event that no matches can be accomplished. Visual and audio feedback is provided for both correct and incorrect moves, as well as level completion. An example of the interface can be seen in Figure 1.

In Tile-o-Scope Grid, levels can be customized by several parameters: the *dataset* (which includes possible images and categories), *grid size*, and *number of tiles of each category to collect*. These parameters can be used to adjust the difficulty of a level. The specific images used in any particular level are chosen at random from the dataset.

For the purposes of this work, we focused on identifying 3 distinct difficulties for levels. The citizen science related datasets were chosen for their application to aerial imagery analysis and identification of features of interest. The following difficulties were chosen:

- Easy (*E*): Matching colors; this difficulty had no citizen science task. Tiles contain images from a Colors dataset, with two possible categories: Red and Blue. The grid was 4×4, requiring collection of 4 tiles from each category.
- Medium (*M*): Looking for tennis courts. Tiles contain geo-located aerial images from near the campus area, sourced using Google Maps. The two possible cate-

	E	M	H
N	53	66	58
# Levels* ** (**E–M***, **E–H***, **M–H***)			
median	24	12	2
# Tiles* (**E–M***, **E–H***)			
median	264	237	90
mean	295	212	261
# Moves* (**E–M***)			
median	61	50	79
Time (s)* ** (**E–M****, **E–H***, **M–H****)			
median	129	212	566
Avg Level Time (s)* **			
			(**E–M***, **E–H***, **M–H***)
median	6	21	151

***:$p < 0.001$;**:$p < 0.01$;*:$p < 0.05$

Table 1: Summary of performance for players per condition for the difficulty validation HIT. **Bold** indicates $p < 0.05$ for omnibus and post-hoc pairwise tests. (X–Y) indicates comparison between *X* and *Y* conditions.

gories were: Tennis Court and No Tennis Court. The grid was 5×5, requiring collection of 8 tiles from each category.

- Hard (*H*): Looking for bridges. Tiles contain geo-located aerial images sourced from a publicly available data set of Civil Air Patrol's aerial images of the 2013 Colorado floods, provided by the U.S. Geological Survey's Hazards Data Distribution System [21]. The two possible categories were: Bridge and No Bridge. The grid was 6×6, requiring collection of 12 tiles from each category.

The 3 difficulties were designed to offer differing cognitive and physical challenges to players [5]. The *E* difficulty uses a dataset with the smallest cognitive challenge, in matching colors, and physical challenge, in grid size and collection requirements, as opposed to the *H* difficulty, whose dataset poses the biggest cognitive challenge, along with increased collection requirements and grid size.

Player Recruitment and Analysis

To recruit players for data analysis, we ran three HITs on Amazon Mechanical Turk, described in more detail below. The first HIT was to validate our difficulty settings; the second was to gather training data to build the Q-table; and the third was to evaluate the Q-learning difficulty ordering. For each HIT, the base rate was $0.10; a bonus of $0.01 for each tile required to complete a level collected was awarded, up to a maximum bonus of $1.90, for a maximum $2.00 payment.

For HITs that compared different difficulty orderings, we randomly assigned each player to a different difficulty ordering condition. We used a Kruskal-Wallis omnibus test to compare numerical metrics. For metrics where the omnibus test came out significant, we then performed post-hoc pairwise comparisons among all conditions, using pairwise Wilcoxon rank sum tests with a Holm correction.

Difficulty Validation

In order to verify that the difficulties were significantly different and to gather information on easy, medium and hard difficulties, we ran a difficulty validation HIT recruiting 150 workers, plus 27 who started but for various reasons did not complete the HIT. Each worker was randomly assigned to play levels of only one difficulty, as many times as they wanted.

Metrics are presented in Table 1. N refers to number of workers in each condition. The results of this HIT indicate that the different difficulties used did, in fact, impact player performance in the game. For example, players completed the most levels of Easy difficulty, and the fewest of Hard difficulty.

Q-learning Formulation and Training

To generate sequences of difficulties, we used a Q-learning based algorithm, a model-free reinforcement learning algorithm [25]. Q-learning is often used in machine learning applications to generate optimal policies based on available data. The algorithm is based on constructing and updating a Q-table of $Q(s_t, a_t)$ values of pairs of states s_t and actions a_t, iteratively updated from training examples using the equation: $Q(s_t, a_t) \leftarrow (1 - \alpha) \cdot Q(s_t, a_t) + \alpha \cdot (r_t + \lambda \cdot \max_a Q(s_{t+1}, a))$, where r_t is the reward of taking action a_t from state s_t, α is the learning rate and λ is the discount factor. Once the Q-table is computed using the available data, it is used to generate the optimal policy.

We represent state s_t as the history of (up to) the last three difficulties encountered by the player in order (e.g. $-$, E, EM, EMM, MMH, etc), with an additional *terminal* state X that represents the player quitting. Possible actions $a_t \in \{E, M, H\}$ can be to serve one of the 3 available difficulties. Taking an action appends that action to the current state, removing the first difficulty from the history if needed,

	Q	G	R
N	55	65	53
# Levels* ** (**Q–G***, **Q–R***, **G–R***)			
median	9	8	12
# Tiles			
median	363	338	305
# Moves* (**Q–R***, **G–R***)			
median	195	218	134
Time (s)* (*Q–R*, *G–R*)			
median	992	1057	624
Avg Level Time (s)*** (**Q–G***, **Q–R***, **G–R***)			
median	115	176	62
Fitness* (**Q–R***, *G–R*)			
median	388	406	280
mean	376	344	267
Avg Move Length (# Tiles)*** (**Q–G***, **Q–R***, **G–R***)			
median	3.1	2.2	3.9

***:$p < 0.001$;**:$p < 0.01$;*:$p < 0.05$

Table 2: Summary of performance for players per condition in the ordering comparison HIT. The *fitness* metric corresponds to the sum of tiles collected of each difficulty, multiplied by the respective difficulty weight. **Bold** indicates $p < 0.05$ and *italics* indicates $p < 0.1$ for omnibus and post-hoc pairwise tests. (X–Y) indicates comparison between *X* and *Y* conditions.

unless that next level was the last level played, in which case the state becomes X (e.g., taking action H from state EMH moves to either X or MHH, depending on whether the player quit during the next level or not, respectively). The reward $r_t = w_{a_t} C_t$ is the number of tiles collected in the next level C_t, weighted by a per-difficulty weight w_{a_t}.

Reward difficulty weights were set to $w_E = 0.0$, $w_M = 1.0$ and $w_H = 1.2$. The E difficulty was given weight 0.0 as the color dataset does not correspond to a meaningful image labeling task; the weights for M and H were set to make the H difficulty more valuable and determined based on the ratio of mean average tiles collected in the difficulty validation HIT. Similarly, the parameters of the algorithm (learning rate $\alpha = 0.001$ and discount factor $\lambda = 0.95$, along with the number of examples to use in training $N = 3 \times 10^5$) were set accordingly so that the rewards for the first action would be proportional to the mean average number of tiles collected in the difficulty validation HIT for that category.

To gather data on player performance to train the Q-table, we ran a subsequent training HIT on Amazon Mechanical Turk, recruiting again 150 workers, plus 20 who started but did not complete the HIT. In this HIT, all workers were each assigned their own (uniform) randomly generated sequence of level difficulties, using the 3 available difficulties. We then drew N examples from the data gathered during this HIT. We picked one random trajectory and one random training example at a time to update the Q-table. This approach let us simulate having a larger dataset.

Evaluation

While the Q-learning algorithm generates an optimal policy by picking the highest valued action for the current state based on the Q-table (i.e. $\mathrm{argmax}_{a_t} Q(s_t, a_t)$), our approach performed a weighted random selection, where each action weight is set as $Q(s_t, a_t)^2$ (i.e the squared value of the action). This allows a small level of exploration, while retaining a strong preference for higher valued actions.

To evaluate our algorithm, we ran an evaluation HIT on Amazon Mechanical Turk, recruiting 150 workers, plus 23 who did not complete the HIT, using the same base rates and bonuses as the previous HITs. Workers were randomly assigned to one of the following conditions:

- *Q-learn* (Q): Sequences generated using the Q-learning based algorithm.
- *Greedy* (G): Sequences that contain only the highest valued difficulty (*H*) based on the weight values.
- *Random* (R): Sequences generated by selecting a difficulty uniform randomly, using the 3 available difficulties. The difference between *Random* and *Q-learn* ordering is in the weights for randomly selecting a difficulty.

A summary of results can be found in Table 2. The fitness metric corresponds to the sum of tiles collected from all categories, multiplied by the relevant weight of that category. For example, if a player collected 10, 20 and 30 tiles from *E*, *M* and *H* categories respectively, the fitness value would be $10 \times 0.0 + 20 \times 1.0 + 30 \times 1.2 = 56$. This metric is an indicator of meaningful labels provided, as it takes into consideration the importance of the level's difficulty.

Our post-hoc pairwise comparison analysis revealed that the *Q-learn* condition outperformed the *Random* condition both in terms of weighted tiles collected, as observed in the fitness metric, as well as the number of moves and total time spent playing the game. While players in the *Random* condition completed the most levels, that can likely be attributed to the presence of more *E* levels than in the other conditions.

Future Work

Our implementation considers only past data for the generation of the Q-table, but can easily be adapted to perform in an online fashion, by considering both past and current data. Similarly, our current approach is not adaptive to individual player performance; however, such information could be incorporated into our formulation. Using a maximum state history size of 3 enabled the direct construction of the Q-table of the Q-learning algorithm, which may not be feasible when larger state spaces are required. This limitation may be tackled by using approximation techniques for the mappings, similar to Chen et al. [3]. This work focused on one game and on specific elements of difficulty as factors for user engagement. Expanding the variety of games tested and the range of factors, along with allowing users to rank their game experience may offer even more insights into the use of this approach in the future.

When comparing *Q-learn* to the *Greedy* approach, which serves only levels of the highest value, i.e *H* difficulty, we found that players completed significantly fewer levels, spent significantly more time per level and attempted moves of significantly smaller length in the *Greedy* condition.

The fitness of the *Q-learn* to the *Greedy* approaches were not found to be significantly different. However, we note that *Q-learn* had the highest mean fitness, and higher early fitness in the retention curve (discussed below), which may indicate directions for further exploration of this approach.

Additionally, of the 3 conditions, *Greedy* offers the least amount of variety, as it essentially repeats the same category. As a result, players in this condition only encounter and contribute to one dataset, providing the least diverse label output. On the other hand, the *Random* and *Q-learn* conditions are able to leverage the game design to combine multiple datasets, which means that players are exposed to different types of labeling tasks and end up contributing to more datasets.

However, the *Q-learn* condition is able to leverage existing data to generate more engaging experiences, as showcased by statistics like time and number of moves, with a higher contribution, as showcased by the significant difference in the fitness metric, when compared to *Random*.

The retention rate of players for fitness across all conditions can be found in Figure 2. Both the *Random* and *Greedy* conditions experience an earlier drop-off in the percent of players above a given fitness, although *Greedy* recovers later. Players in the *Random* condition end up contributing significantly less, while the behavior of players in the *Greedy* condition is in line with findings in the literature that indicate that most participants quit early on, with a smaller subset ending up contributing bulk of the work. In contrast,

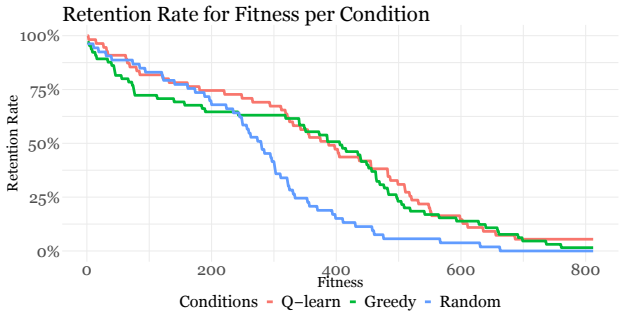

Figure 2: Retention rate of players over fitness.

the *Q-learn* condition is, visually, able to provide a smoother drop-off of participants, suggesting a potentially better approach for retaining engagement in the task.

Conclusion

We presented a web game for image labelling called Tile-o-Scope Grid, which combines gamification elements with reinforcement learning, by providing automated difficulty sequence generation using an approach based on Q-learning. Our preliminary results indicate potential towards the use of algorithms such as Q-learning for designing engaging citizen science games and web interfaces for image labeling, that are able to seamlessly combine different datasets in an optimal fashion. Tile-o-Scope Grid is part of a research project called Cartoscope, which can be found at http://cartosco.pe.

Acknowledgments

We thank all the players. This material is based upon work supported by the National Science Foundation under grant no. 1816426.

REFERENCES

1. Luke Barrington, Shubharoop Ghosh, Marjorie Greene, Shay Har-Noy, Jay Berger, Stuart Gill, Albert Yu-Min Lin, and Charles Huyck. 2012. Crowdsourcing earthquake damage assessment using remote sensing imagery. *Annals of Geophysics* 54, 6 (Jan. 2012). DOI: http://dx.doi.org/10.4401/ag-5324
2. Carrie J. Cai, Shamsi T. Iqbal, and Jaime Teevan. 2016. Chain reactions: the impact of order on microtask chains. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.
3. Zhengxing Chen, Christopher Amato, Truong-Huy D. Nguyen, Seth Cooper, Yizhou Sun, and Magy Seif El-Nasr. 2018. Q-DeckRec: a fast deck recommendation system for collectible card games. In *2018 IEEE Conference on Computational Intelligence and Games*. 1–8.
4. Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović, and Foldit players. 2010. Predicting protein structures with a multiplayer online game. *Nature* 466, 7307 (2010), 756.
5. Alena Denisova, Christian Guckelsberger, and David Zendle. 2017. Challenge in digital games: towards developing a measurement tool. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. 2511–2519. DOI: http://dx.doi.org/10.1145/3027063.3053209
6. J. Richard Hackman and Greg R. Oldham. 1976. Motivation through the design of work: test of a theory. *Organizational Behavior and Human Performance* 16, 2 (Aug. 1976), 250–279. DOI: http://dx.doi.org/10.1016/0030-5073(76)90016-7
7. Robin Hunicke. 2005. The case for dynamic difficulty adjustment in games. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*. 429–433.
8. Aniket Kittur, Jeffrey V. Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. 2013. The future of crowd work. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*. 1301–1318. DOI:http://dx.doi.org/10.1145/2441776.2441923
9. Walter S. Lasecki, Adam Marcus, Jeffrey M. Rzeszotarski, and Jeffrey P. Bigham. 2014. *Using microtask continuity to improve crowdsourcing.* Technical Report CMU-HCII-14-100. School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania. http://reports-archive.adm.cs.cmu.edu/anon/hcii/CMU-HCII-14-100.pdf
10. Travis Mandel, Yun-En Liu, Emma Brunskill, and Zoran Popović. 2016. Offline evaluation of online reinforcement learning algorithms. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. 1926–1933.
11. Travis Mandel, Yun-En Liu, Sergey Levine, Emma Brunskill, and Zoran Popovic. 2014. Offline policy evaluation across representations with applications to educational games. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*. 1077–1084.
12. Robert Munro, Tyler Schnoebelen, and Schuyler Erle. 2013. Quality analysis after action report for the crowdsourced aerial imagery assessment following Hurricane Sandy. In *Proceedings of the 10th*

International Conference on Information Systems for Crisis Response and Management.

13. Ferda Ofli, Patrick Meier, Muhammad Imran, Carlos Castillo, Devis Tuia, Nicolas Rey, Julien Briant, Pauline Millet, Friedrich Reinhard, Matthew Parkan, and Stéphane Joost. 2016. Combining human computing and machine learning to make sense of big (aerial) data for disaster response. *Big Data* 4, 1 (Feb. 2016), 47–59. DOI:http://dx.doi.org/10.1089/big.2014.0064

14. Playdots, Inc. 2013. *Dots*. Game [Mobile]. (30 April 2013). Playdots, Inc., New York City, USA.

15. M. Jordan Raddick, Georgia Bracey, Pamela L. Gay, Chris J. Lintott, Phil Murray, Kevin Schawinski, Alexander S. Szalay, and Jan Vandenberg. 2010. Galaxy Zoo: exploring the motivations of citizen science volunteers. *Astronomy Education Review* 9, 1 (Dec. 2010). DOI: http://dx.doi.org/10.3847/AER2009036 arXiv: 0909.2925.

16. Sofia Eleni Spatharioti and Seth Cooper. 2017. On variety, complexity, and engagement in crowdsourced disaster response tasks. In *Proceedings of the 14th International Conference on Information Systems for Crisis Response And Management*. 489–498.

17. Sofia Eleni Spatharioti, Rebecca Govoni, Jennifer S. Carrera, Sara Wylie, and Seth Cooper. 2017. A required work payment scheme for crowdsourced disaster response: worker performance and motivations. In *Proceedings of the 14th International Conference on Information Systems for Crisis Response And Management*. 475–488.

18. Tobias Sturn, Michael Wimmer, Carl Salk, Christoph Perger, Linda See, and Steffen Fritz. 2015. Cropland Capture – a game for improving global cropland maps. In *Proceedings of the 10th International Conference on the Foundations of Digital Games*.

19. Alexandra Burchard Swanson. 2014. *Living with lions: spatiotemporal aspects of coexistence in savanna carnivores*. Ph.D. dissertation. University of Minnesota. http://conservancy.umn.edu/handle/11299/167642

20. Chek Tien Tan, Daniel Rosser, and Natalie Harrold. 2013. Crowdsourcing facial expressions using popular gameplay. In *SIGGRAPH Asia 2013 Technical Briefs*. Article 26, 4 pages. DOI: http://dx.doi.org/10.1145/2542355.2542388

21. U.S. Geological Survey. 2019. Hazards Data Distribution System Explorer. (2019). http://hddsexplorer.usgs.gov/

22. Luis von Ahn and Laura Dabbish. 2004. Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Vienna, Austria, 319–326. DOI: http://dx.doi.org/10.1145/985692.985733

23. Luis von Ahn, Ruoran Liu, and Manuel Blum. 2006. Peekaboom: a game for locating objects in images. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 55–64. DOI: http://dx.doi.org/10.1145/1124772.1124782

24. Jeffrey Yoo Warren. 2010. *Grassroots mapping: tools for participatory and activist cartography*. Thesis. Massachusetts Institute of Technology. http://dspace.mit.edu/handle/1721.1/65319

25. Christopher John Cornish Hellaby Watkins. 1989. *Learning from delayed rewards*. Ph.D. Dissertation. King's College, Cambridge.