Interactive Visualizer to Facilitate Game Designers in Understanding Machine Learning

Jiachi Xie
Drexel University
Philadelphia, PA, USA
xieicecream@gmail.com

Chelsea M. Myers
Drexel University
Philadelphia, PA, USA
chel.myers@gmail.com

Jichen Zhu
Drexel University
Philadelphia, PA, USA
jichen.zhu@gmail.com

ABSTRACT

Machine Learning (ML) is a useful tool for modern game designers but often requires a technical background to understand. This gap of knowledge can intimidate less technical game designers from employing ML techniques to evaluate designs or incorporate ML into game mechanics. Our research aims to bridge this gap by exploring interactive visualizations as a way to introduce ML principles to game designers. We have developed *QUBE*, an interactive level designer that shifts ML education into the context of game design. We present *QUBE's* interactive visualization techniques and evaluation through two expert panels (n=4, n=6) with game design, ML, and user experience experts.

KEYWORDS

machine learning, interactive visualizations, game design

CHI'19 Extended Abstracts, May 4–9, 2019, Glasgow, Scotland Uk

ACM ISBN 978-1-4503-5971-9/19/05.
https://doi.org/10.1145/3290607.3312851

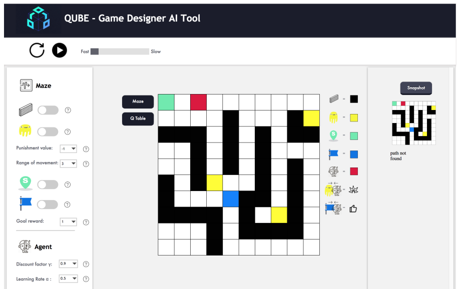

Figure 1: Screen shot of the original prototype of *QUBE* and its *maze* view of the level. Visit at latest version at http://jiachixie.com/Qube/qube.html

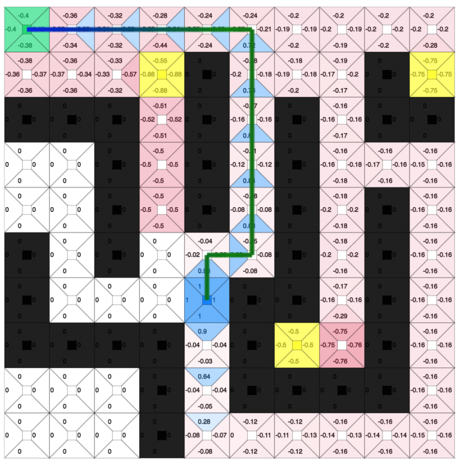

Figure 2: Screen shot of the *QUBE* tool and its Q-Tabe view of the level

INTRODUCTION

Machine Learning (ML) can be an extremely useful tool to game designers and allows them to accomplish a variety of techniques such as adaptive gameplay and the dynamic generation of levels. However, game designers might be intimidated by ML since its explanation relies heavily on mathematical knowledge; area game designers may not be familiar with. Because of this gap of tools and knowledge, game designers may miss new design opportunities afforded by ML.

We aim to bridge this gap by exploring the approach of interactive visualization. In particular, we created an interactive visualizer that introduces game designers to ML by leveraging the familiar act of level design. Our research employs a user-centered design approach to iteratively build this visualizer and evaluate its effectiveness. We have built *QUBE* (Fig. 1), which introduces game designers to *reinforcement learning*, a subset of ML. *QUBE* is primarily a ML educational tool. Its interface allows users to edit the factors of a *Q-Learning* algorithm, a value-based reinforcement learning algorithm, powering an agent moving through a level with enemies.

Interactive visualizers for ML focus primarily on fostering collaboration with ML or helping users analyze ML results [4, 12, 14]. ML interactive visualizers, such as Google's TensorFlow Playground [1], PathFinding.js [15], and GAN Lab [5] succeed in creating engaging visualizations. However, these visualizers often do not explain the underlying processes powering the visuals or still rely on heavy ML jargon making the visualizers not accessible to game designers with limited computer science backgrounds. Additionally, the on-boarding necessary to educate users on how to operate these tools exists as text-based tutorials displayed under the tool. With *QUBE*, we evaluate this tutorial method initially and explore alternative techniques for ML education for game designers.

By shifting ML education into the context of something familiar to game designers (e.g., level design) we aim to demystify ML. We have reviewed and edited *QUBE's* design through two expert panels (n=4, n=6) with a range of ML, game design, and user experience (UX) experts. In this paper, we present *QUBE's* design, our expert panel results, and our next steps to evaluate *QUBE* further. We also present our proposed techniques for interactive visualizations for ML education and the success of these techniques with our experts.

RELATED WORK

ML is a valuable tool for game design through various applications [2, 7–9, 11, 13, 17]. For a majority of ML interactive visualizers, foundational knowledge of ML is required to understand how the ML processes work and to use the tools. Educational tools such as Google's TensorFlow Playground [1] and Pathfinding.js [15] visualize neural networks and pathfinding algorithms respectively, but do not explain the principles of the processes. These tools use techniques such as removing the need to code, focusing on a specific domain, and limiting user input to make ML more approachable [16].

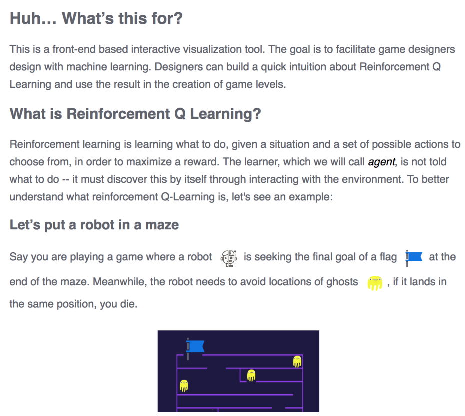

Figure 3: Screen shot of the *QUBE* tool's original tutorial

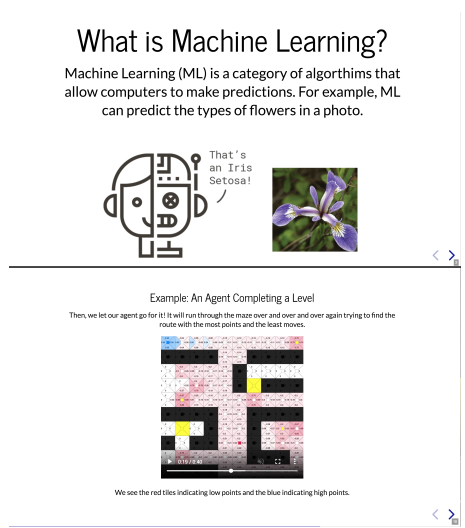

Figure 4: Samples slides for *QUBE's* new tutorial after EP1 results.

GAN Lab's [5] interactive visualization techniques, such as their *step-by-step* view, breaks down the Generative Adversarial Network (GAN) process in an attempt to improve comprehension. However, GAN Lab's user interface includes heavy ML jargon. For GAN Lab and TensorFlow users must learn these ML terms through a text-based tutorial displayed under the tool. For a game designer who may not even know what "epoch" refers to, this method of on-boarding may be overwhelming. With *QUBE*, we explore additional techniques to better educate ML terms and processes.

To our knowledge, little research exists on how to educate game designers on ML principles. Yang et. al. [16] interviewed general non-experts on how they utilized ML interactive visualizers and presented several pitfalls these users are susceptible too. Zhu et. al. [18] propose a new area of research, *eXplainable AI for Designers,* which focuses on creating explainable AI techniques specifically for game designers (e.g., highlighting transparent and collaborative techniques). We hypothesize that an interactive visualizer could also be an engaging tool for game designers to learn ML. However, current ML interactive visualizers still remain too complex for easy comprehension. Outside of game design, interactive visualizers have been found to provide support in learning activities in online learning managers [6] and in computer science education [10]. *QUBE* builds upon the successes of previous interactive visualizers as we analyze the needs of a new user group; game designers.

QUBE'S DESIGN

Our first design of *QUBE* sought to build on previous ML interactive visualizer techniques and evaluate their effectiveness with game designers. *QUBE* is a web-based tool designed to shift the context of learning ML to a game design setting to promote active learning. Active learning techniques are commonly characterized as interactive, simple to understand, quick to complete, creative, and relevant [3, 10]. When designing *QUBE,* we developed techniques that meet these characteristics but specifically for the game design audience. For example, our ML educational content was structured to not overwhelm game designers and focuses on *Q-Learning* to keep the educational content short. *QUBE* was also designed to be an interactive basic level designer to keep ML education simple, creative, and relevant to game designers.

The goal of *QUBE* is to facilitate the comprehension of Q-Learning and the factors that augment the results of the algorithm. To evaluate the on-boarding approach seen in previous ML interactive visualizers [1, 5] a majority of our educational content is under the level designer (Fig. 3). Ideally, a user would first read through the educational content and then interact with the level designer. In the level designer, users are given a 10x10 grid where they can design Pac-Man style levels with a Q-Learning-powered agent cycling through the level and calculating the best path through it. Users can alter this path by drawing walls to build mazes, adding ghosts as enemies, and changing the start and end-goal of the agent. Users can see how the agent processes the level through two visualization techniques; the *maze visualization* (Fig. 1) and *Q-Table visualization* (close up in Fig. 2). After the

Expert Panel I (EP1) Participants	
EP1-1	Computer science faculty and game designer
EP1-2	UX researcher
EP1-3	Game designer
EP1-4	Game designer and digital media PhD Student
Expert Panel II (EP2) Participants	
EP2-1	Game designer and digital media PhD Student
EP2-2	Game ML researcher
EP2-3	Computer scientist and game developer
EP2-4	UI/UX designer
EP2-5	Data scientist with ML expertise
EP2-6	UX researcher

Table 1: Experts recruited and their expertise for Expert Panel I & II. Experts were NOT re-recruited for EP2.

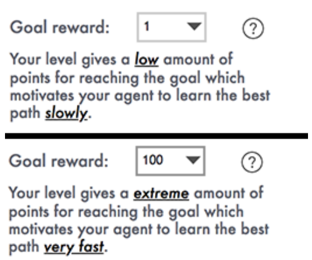

Figure 5: Example of the *mad lib explainer* implemented after EP1 results for the Goal Reward factor with a value of 1 and 100 respectively.

user designs a level she can click "play" (seen above the level). Both visualizations show the agent cycling through the level, encountering ghosts and "dying," and finding the goal and being "rewarded". However, the Q-Table visualization attempts to make the Q-Learning process transparent and overlays the algorithm's Q-Table over the level to show the agent's assessment on which paths to take. For both visualizations, once the agent is finished its assessment, the optimal path calculated is displayed over the level (as seen in Fig. 2) for the user to view.

In the left sidebar (Fig. 1), users can edit the agent's Q-Learning factors to help illustrate that the algorithm can be customized to yield different results and paths. Tooltips are present and define each factor. We felt that this connection to the educational content was missing from previous ML interactive visualizers and would allow users to find definitions of ML terms without searching tutorial content. Finally, the snapshot feature takes a snapshot of the level design and allows users to reflect on previous levels and compare; a valuable feature to observe the effect of the factors.

EXPERT PANEL I

In the Expert Panel I (EP1), we recruited four experts in separate hour-long usability tests of *QUBE* from academia and industry. We recruited both game and UX experts. We first gave a brief introduction to *QUBE* and then allowed our expert to explore *QUBE* while using the think aloud protocol. After, we interviewed each expert focusing on *QUBE's* educational effect and usability.

Results — We refer to our experts in this analysis as EP1-#, where EP1 is the expert panel number and '#' is the number of the participant. The panel's expertise can be seen in Table 1. All experts believed both visualizations of the agent's process would effectively help designers understand Q-Learning. However, although a general mental model of Q-Learning could accurately be formed, experts (EP1-2, EP1-3) were still unclear on the impact of some factors. We observed this confusion was caused by the dense educational content under the tool and its disconnect from our level designer.

Implemented Improvements — To address our findings, we reorganized the educational content for *QUBE*. We decided to break up the educational content into tutorial slides and re-hauled the text to present each factor independently (Fig. 4). We also designed a new technique we call *mad lib explainers*, Fig. 5, which embeds adaptive educational content next to each factor. The mad lib explainers are text shown under each factor in *QUBE's* sidebar with certain words that swap to correspond with the selected value to illustrate its impact. We believe this technique illustrates how the factors impact the agent without running the simulation over again. We hypothesize this technique can help lessen the cognitive load of remembering what factors were changed and yielded different outcomes.

EXPERT PANEL II

After implementing improvements, we conducted a second expert panel. Our Expert Panel II (EP2) sessions were structured the same as EP1.

Results — In EP2, we recruited six experts to review *QUBE* (Table 1). We found again the visualizations of the agent cycling through the level informed a correct general mental model of how the Q-Learning algorithm worked for all our experts. EP2-1, EP2-2, EP2-3, and EP2-6 believed that our mad lib explainers were successful in illustrating the impact of changing each factor's value. Additionally, we found that while our educational content was improved through the new tutorial slide format, it still was not engaging enough for our less technically-savvy experts (EP2-2, EP2-4, EP2-6). Too much time elapsed between the presentation of the Q-Learning factors and the ability to manipulate them, causing the experts to forget the factors' impact. This caused our experts to avoid manipulating the more complex factors (e.g., Discount Factor). However, our more technically-savvy experts desired more technical details (EP2-2, EP2-3, EP2-5). These results highlight a discrepancy in our experts' opinions as some call for more technical explanations while others less.

DISCUSSION & CONCLUSION

QUBE's primary focus, to educate game designers on ML principles through Q-Learning, can still be strengthened. We found that the common approach of placing educational content under the interactive visualizer was not found effective by our experts. EP2 revealed that while our mad lib explainers were successful in better connecting educational content to our interactive visualizations, the tutorial slides were still too dense or not engaging for our less technically-savvy experts. Since these experts are the closest to our intended users of *QUBE*, this poses as our biggest challenge. Based on our results, we propose that the tutorial slides allows users to actually manipulate the factors through our mad lib explainers. We speculate that by changing the slides to allow factor customization, users will be able to observe the effects of each factor in the slide. This also introduces our mad lib explainers earlier in our users' experience with *QUBE*. This hopefully addresses the issue of experts forgetting factors due to the time elapsed between the tutorial slides and visualizer since the mad lib explainers can serve as an interactive visual connecting the two segments of *QUBE*.

Our next steps are to incorporate these changes and evaluate *QUBE* in a user study with game design students. Our goal is to create an interactive visualization tool to demystify ML for game designers by introducing them to reinforcement learning, specifically Q-Learning. With our current tool, *QUBE*, we have conducted two expert panels to evaluate its design and educational effect. We found in addition to general usability improvements, a standard tutorial approach was not engaging enough to teach non-technically savvy experts the factors impacting the Q-Learning algorithm. We did, however, receive feedback that *QUBE's* visualization methods do make how Q-Learning works in general transparent to our experts. We believe while our level design tool approach is not generalizable to all ML educational topics, our proposed mad lib explainers are domain independent. We will continue improving *QUBE* through revisions and future studies.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* (2016).

[2] Fernando de Mesentier Silva, Scott Lee, Julian Togelius, and Andy Nealen. 2017. AI-based playtesting of contemporary board games. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*. ACM, 13.

[3] Jennifer L Faust and Donald R Paulson. 1998. Active learning in the college classroom. *Journal on excellence in college teaching* 9, 2 (1998), 3–24.

[4] Geoffrey Holmes, Andrew Donkin, and Ian H Witten. 1994. Weka: A machine learning workbench. In *Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on*. IEEE, 357–361.

[5] Minsuk Kahng, Nikhil Thorat, Duen Horng, Polo Chau, Fernanda B Vi, and H C Sep. 2019. GAN Lab : Understanding Complex Deep Generative Models using Interactive Visual Experimentation. 25, 1 (2019). arXiv:arXiv:1809.01587v1

[6] Kirsi Kuosa, Damiano Distante, Anne Tervakari, Luigi Cerulo, Alejandro Fernández, Juho Koro, and Meri Kailanto. 2016. Interactive visualization tools to improve learning and teaching in online learning environments. *International Journal of Distance Education Technologies (IJDET)* 14, 1 (2016), 1–21.

[7] Lars Lidén. 2003. Artificial stupidity: The art of intentional mistakes. *AI game programming wisdom* 2 (2003), 41–48.

[8] Jialin Liu, Julian Togelius, Diego Pérez-Liébana, and Simon M Lucas. 2017. Evolving game skill-depth using general video game ai agents. In *Evolutionary Computation (CEC), 2017 IEEE Congress on*. IEEE, 2299–2307.

[9] Jorge Maturana, Álvaro Fialho, Frédéric Saubion, Marc Schoenauer, and Michèle Sebag. 2009. Extreme compass and dynamic multi-armed bandits for adaptive operator selection. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*. IEEE, 365–372.

[10] Dino Schweitzer and Wayne Brown. 2007. Interactive visualization for the active learning classroom. *ACM SIGCSE Bulletin* 39, 1 (2007), 208–212.

[11] Adam M Smith, Mark J Nelson, and Michael Mateas. 2009. Computational Support for Play Testing Game Sketches. In *AIIDE*.

[12] Justin Talbot, Bongshin Lee, Ashish Kapoor, and Desney S Tan. 2009. EnsembleMatrix: interactive visualization to support machine learning with multiple classifiers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1283–1292.

[13] Julian Togelius, Noor Shaker, and Mark J Nelson. 2014. Procedural Content Generation in Games: A Textbook and an Overview of Current Research/J. *Togelius, N. Shaker, M. Nelson-Berlin: Springer* (2014).

[14] Malcolm Ware, Eibe Frank, Geoffrey Holmes, Mark Hall, and Ian H Witten. 2001. Interactive machine learning: letting users build classifiers. *International Journal of Human-Computer Studies* 55, 3 (2001), 281–292.

[15] Xueqiao Xu. [n. d.]. A comprehensive path-finding library in javascript. http://qiao.github.io/PathFinding.js/visual/.

[16] Qian Yang, Jina Suh, Nan-Chen Chen, and Gonzalo Ramos. [n. d.]. ([n. d.]).

[17] Georgios N Yannakakis and Julian Togelius. 2017. *Artificial Intelligence and Games*.

[18] Jichen Zhu, Antonios Liapis, Sebastian Risi, Rafael Bidarra, and G Michael Youngblood. 2018. Explainable AI for designers: A human-centered perspective on mixed-initiative co-creation. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, 1–8.