

Repurposing Citizen Science Games as Software Tools for Professional Scientists

Seth Cooper
Northeastern University
scooper@ccs.neu.edu

Amy L. R. Sterling
Princeton University
amy@eyewire.org

Robert Kleffner
Northeastern University
rob.kleffner@gmail.com

William M. Silversmith
Princeton University
will@eyewire.org

Justin B. Siegel
University of California, Davis
jbsiegel@ucdavis.edu

ABSTRACT

Scientific software is often developed with *professional* scientists in mind, resulting in complex tools with a steep learning curve. *Citizen science games*, however, are designed for *citizen* scientists—members of the general public. These games maintain scientific accuracy while placing design goals such as usability and enjoyment at the forefront. In this paper, we identify an emerging *use of game-based technology*, in the repurposing of citizen science games to be software tools for professional scientists in their work. We discuss our experience in two such repurposings: Foldit, a protein folding and design game, and Eyewire, a web-based 3D neuron reconstruction game. Based on this experience, we provide evidence that the *software artifacts* produced for citizen science can be useful for professional scientists, and provide an overview of key design principles we found to be useful in the process of repurposing.

CCS CONCEPTS

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

KEYWORDS

citizen science; games; design; repurposing

ACM Reference Format:

Seth Cooper, Amy L. R. Sterling, Robert Kleffner, William M. Silversmith, and Justin B. Siegel. 2018. Repurposing Citizen Science Games as Software Tools for Professional Scientists. In *Foundations of Digital Games 2018 (FDG18)*, August 7–10, 2018, Malmö, Sweden. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3235765.3235770>

1 INTRODUCTION

In the early 2000s, a new type of software began to emerge, in the hopes of procuring insights from a growing ocean of data: online citizen science. Though citizen science has a history of over 100 years, online platforms are a recent emergence. It was with this emergence that citizen science began to draw inspiration from the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
FDG'18, August 7–10, 2018, Malmö, Sweden

© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-6571-0/18/08...\$15.00
<https://doi.org/10.1145/3235765.3235770>

world of computer gaming, with the goal of turning novices into experts and retaining users. The reorganization of scientific research from professional scientists (e.g., researchers or students with formal training in a field) to citizen scientists (e.g., members of the general public who may not have formal training) necessitates considerations in software design that are largely considered irrelevant in standard scientific software development. Scientific software produced for citizen science often prioritizes usability and enjoyment. While data gathering and analysis are typically considered to be the benefits of citizen science, we observe that, in addition, the software artifacts produced for citizen science themselves can be useful for professional scientists.

In this paper we discuss our experience with two *citizen science game* projects that independently followed a similar development trajectory: originally developed for citizen scientists, the games were later repurposed to be used by professional scientists directly in their work. **Foldit**, a game in which players predict the structures of naturally-occurring proteins and design synthetic proteins [8] was restructured into a separate stand-alone, offline application for protein structure manipulation, called **Foldit Standalone** [26]. **Eyewire**, a game in which players decipher the 3D structure of neurons by mapping them across z-stacks of images [25], was extended to allow professional scientists to load their own data sets into the game, referred to in this paper as the **Eyewire Tracer Interface**.

This work makes the following contributions: (i) we identify a newly-emerging *use of game-based technology* in the repurposing of citizen science games as software tools for professional scientists, demonstrating that the software artifacts (and not just the data) produced by citizen science games can be of use to professional scientists; and (ii) we identify key design principles from our repurposing experiences. We believe this work presents a starting point for possible further study of this practice.

In Section 2, we provide background and related work. In Sections 3 and 4, we discuss the history of the two games and their repurposing, along with their use by professional scientists. In Section 5 we list the key design principles, and conclude in Section 6.

2 RELATED WORK

This work relates broadly to the areas of *citizen science* and the *use of game-based technology*.

Citizen science allows the general public to participate in scientific research by doing tasks that teams of professional scientists

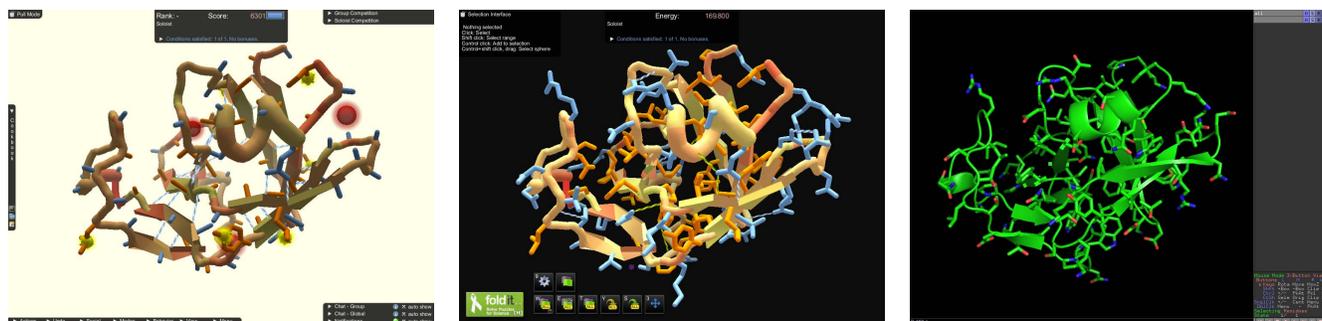


Figure 1: Screenshots of (left) Foldit game default visualization, (center) Foldit Standalone default visualization and (right) PyMOL [11]. All programs have loaded the same protein structure.

would typically do. Early online projects primarily allowed the public to collect environmental data, such as the presence of species in specific ecosystems [42]. The ESP Game had players provide labels for general images [46]. Stardust@home became the first citizen science project to challenge novice participants with the analysis of existing scientific data [48]. In the years that immediately followed, many projects launched. Some succeeded, others stagnated; citizen science growth and retention remains an unsolved problem [15]. Thus researchers grew keen to explore new interaction models in the hopes of improving the success of these new crowdsourcing platforms. GalaxyZoo [2], the debut project of Zooniverse, launched in 2007 with a new style of user-friendly interface and in the first two years recruited over 200,000 participants [34]. While GalaxyZoo successfully recruited many participants, other projects have explored turning into games as a practical means by which to address the challenge of recruiting and retaining users [18], as evidenced by the numerous successful examples of citizen science projects manifesting as serious games. Phylo [23] and Fraxinus [35] both deployed in the genomics field; EteRNA [29] focused on the design of secondary RNA structure; and Nanocrafter [3] examined DNA nanotechnology.

As games are often at the forefront of technology, there are many examples of the *use of game-based technology* for other purposes. This can range from the use of game hardware, such as controllers [39] and other input devices [27], to the use of game software, such as using game engines, for the development of general interactive software and simulations [32, 40, 47]. One class of scientific games, *biotic games*, even aims to advance scientific lab equipment by using it as game hardware [36]. It has become increasingly popular to repurpose existing games for educational purposes (examples can be found in SimCityEDU [17], Minecraft: Education Edition [31], and Learn With Portals [44]). One distinct property of Foldit Standalone and the Eyewire Tracer Interface among other uses of game-based technology is that their gameplay can be used to identify solutions to specific instances of scientific problems.

More broadly, previous work has also examined the (re)appropriation of digital artifacts [4, 12], where users adapt software in ways designers did not intend. In the current work, the original designers were involved in modifying the software for a new group of users (professional scientists), and thus we have used the term “repurposing”. One notable scientific application of existing game

hardware is the Folding@home project. Folding@home, a “screen-saver” volunteer computing project, applied the processing power of unused PlayStation 3s toward simulating protein folding at a massive scale [30]. In some cases, existing games have been adapted to provide new interfaces for work, such as using Doom to manage system processes on a computer [7].

3 GAMES AND REPURPOSING

Foldit is an online game, implemented in cross-platform C++, in which players interactively manipulate protein structures [8, 9]. The game requires downloading and installation to play. Since the game’s release in 2008, Foldit players have contributed to a variety of areas in computational structural biochemistry research [13, 16, 21, 24]. Foldit presents a graphical, interactive front-end to the powerful Rosetta Molecular Modeling Suite [28, 37], which handles the game’s underlying protein model manipulations and scoring. Use of Rosetta was intended to ensure that the protein structures created by players would be scientifically useful.

As Rosetta is a popular tool for protein modeling by professional scientists, it soon became clear that Foldit’s interface could be useful as an interface for users of Rosetta. Thus, development of a version of the game that could be used by professional scientists was started. Foldit Standalone was developed as a separate application from Foldit, although they shared most of the same codebase. The primary differences between Foldit Standalone and Foldit are (i) removing game-related elements such as leaderboards, chat, player accounts, and website server integration; (ii) allowing users to import and export their own data files, rather than being limited to those provided by the game’s server; (iii) modifying the interface’s terminology and default options to be more standard (for example, replacing “score” with “energy” and replacing the game term “wiggle” with the more standard “minimize”); (iv) reporting the protein’s energy calculated by Rosetta directly (lower is better) rather than converting it into a game score (higher is better); and (v) allowing some advanced features that “break game rules” (such as changing parts of the protein that are meant to be frozen). Beyond this, the Foldit and Foldit Standalone interfaces share much in common. The design of Foldit’s interface is discussed in more detail in previous work [9].

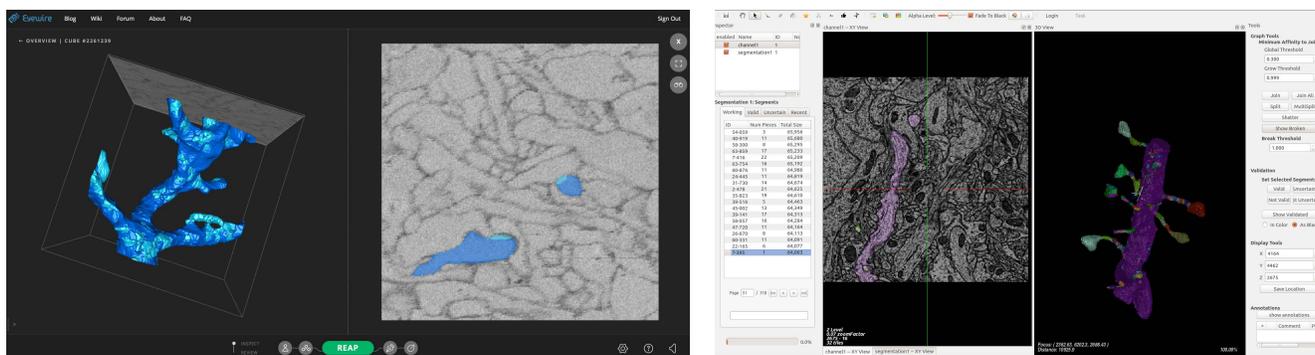


Figure 2: Screenshots of (left) checking a cube in Eyewire and (right) reconstruction in Omni [38].

Eyewire is a web-based online game where players reconstruct 3D neurons. In the late 2000s, the Seung Lab created a C++ 3D image viewer and reconstruction tool called Omni [38]. As a desktop application, it was able to use the full power of whatever hardware it ran on to display immense volumetric images (“volumes”, which are broken down into smaller sections called “cubes”). Tools were added to allow “Tracers”—trained neuroscience reconstruction expert research assistants employed by the lab—to trace neuron branches (“neurites”) on their local machines. Two Tracers would independently reconstruct a section of neurite, and a third would compare their work. Tracers would work from relatively powerful workstations and access required a volume to be loaded on a local drive or a LAN connected NFS mount to achieve acceptable performance. Over the years, the codebase became increasingly inflexible with complex class hierarchies.

The Eyewire platform was originally created to serve neuron image stacks and segmentation to citizen scientist players to aid the reconstruction effort. Eyewire players can win badges and land on a leaderboard by winning points for accurately mapping cubes. Over the years, players improved and Eyewire developed a ranking system whereby each class of player unlocked more advanced features that gave them more power in the game. In parallel to the player interface, features for “Gamemasters”—Eyewire employees who oversee the community and reconstruction—were necessary to streamline the checking process carried out on each cell. The Eyewire Tracer Interface repurposed the Gamemaster controls for use by the trained Tracers, thus bypassing the crowd.

4 USE BY PROFESSIONAL SCIENTISTS

Foldit Standalone has had a version available since 2011, free for academic use. As of early 2018, it has been academically licensed over 1300 times from over 40 different countries. An early prototype of Foldit Standalone was used to iterate on enzymes designed by Foldit players [13]. While still in development, Foldit Standalone had already begun to play a critical role in both research and education [5, 41]. Students have used this tool to re-engineer the reaction specificity of proteases for the development of therapeutics for Anthrax [49] and Celiac Disease [19]. Foldit Standalone has been particularly successful in engaging undergraduate students within the context of the International Genetically Engineering Competition (iGEM). While not all iGEM teams utilize Foldit Standalone,

it has been used by a subset of teams almost every year since its widespread introduction. This is illustrated by the 2011 University of Washington and 2014 University of California, Davis teams, both of which won the Grand Prize at the international competition and used Foldit Standalone as an integral part of their project.

When considering more standard tools, there are a wide range of GUI tools for molecular modeling and visualization of proteins now available, each of which is tailored for a targeted user. In the case of Coot [14] and PHENIX [1], the interfaces have been optimized for use by trained crystallographers as part of a workflow for converting biophysical data (e.g. X-ray diffraction data) into a three-dimensional structure. The widely popular PyMOL [11] and Chimera [33] are optimized for visualization by trained scientists as part of a workflow in evaluating the molecular details of a protein’s structure for obtaining insight into its biological function. Foldit—and thus Foldit Standalone—has been optimized for allowing the user to modulate the shape and composition of a protein’s structure within the Rosetta energy function, either for predicting its naturally occurring structure or engineering a novel one. Though other software has been developed for protein structure manipulation (including ProteinShop [10] and Sculpt [43]), Foldit was designed with novice users at the forefront, and to provide an interface to the protein modeling features of the Rosetta software. Foldit’s primary coloring scheme, for example, is based on the per-residue energy of the protein.

This contrast between PyMOL and Foldit Standalone, for example, can clearly be observed in Figure 1. Here, the same enzyme-ligand complex has been loaded and oriented in the same way in each program. In PyMOL all of the readily accessible options are for fine-tuning the visualization parameters. In Foldit Standalone, all of the readily accessible options are for manipulating structure and providing real-time feedback on the predicted change in energy due to a given manipulation. While both programs have structure manipulation and visualization capabilities, each program has a different emphasis and usage audience in mind.

The **Eyewire Tracer Interface’s** new hypersquare and Msty features (also available in Eyewire) were in development since mid-2016 and a pre-release beta was made available in December 2016 for use by staff Tracers on a zebrafish dataset. A result, showing evidence of synapses between zebrafish integrator neurons, was published by Vishwanathan et al. [45]. This work included cells

reconstructed by professional scientists using the Eyewire Tracer Interface.

In comparison to more standard tools, Eyewire and the scientific software Omni (shown in Figure 2) both begin with VAST (Volume Annotation and Segmentation Tool) [22], which is used to generate ground truth for machine learning algorithms. The output is segmentation used for 3D reconstruction. Various other tools exist to facilitate reconstruction from z-stack electron microscopy datasets. KNOSSOS [20] enables skeletonization, a method of tracing by drawing a line through neurites. Since KNOSSOS does not use AI segmentation, no real-time 3D models are shown, which can hinder tracing through narrow or otherwise difficult branches.

Omni's controls are on par with Eyewire's. However, according to discussion with Tracers, Omni's are more frustrating to use. For example, many Omni keyboard shortcuts require 3 keys and are not explicitly introduced in the interface. There are at a minimum 40 reconstruction-related buttons and options in the Omni interface, compared to 10 in Eyewire. Omni works offline and large volumes must be locally downloaded. Eyewire runs through a browser window where local cubes generated from a larger dataset are downloaded when a user starts playing or inspects a cube. Eyewire enables real-time collaboration through chat and focused coordination through private messages and toolbars.

5 KEY DESIGN PRINCIPLES

Based on our experience repurposing two citizen science games for use by professional scientists, we offer the following key design principles we found to be useful in the repurposing process. These principles relate the design of the interfaces, the software architecture, and game systems. These may be of use for other citizen science games that wish to expand their interface for use by professional scientists, or developers creating scientific software. These key design principles were derived from open-ended discussions between the project teams, reflecting on the design and development processes of the two projects. We do not claim to have created these design principles, as they are commonly found in software development projects. However, we have identified these as key principles that both projects found useful, and thus may generalize to other projects. We believe they are a starting point that should be studied further.

5.1 Familiarity

Leveraging existing interface paradigms makes it easier for those who are used to existing standard scientific software to get started.

Foldit Standalone approaches familiarity largely by building visualization options, mouse controls, and other interface elements suggested by professional scientists, and making those the default configuration. This also had the benefit of making these options available to players of the game version. Thus, the vast majority of visualization options are available to both players of Foldit and users of Foldit Standalone; however, the default settings used by each application are not the same. Foldit's default visualization options are set intentionally to be quite different than many standard protein visualizations, with bright colors and hidden complexity. Foldit standalone, on the other hand, defaults to settings familiar for professional scientists use. Based on discussions with users of

standard protein visualization software, we added options and features that they were familiar with, including a black background, orthographic camera view, specific camera rotation mouse controls, additional geometry such as sticks and spheres, clipping planes, and polar hydrogen visualizations. These options then eventually became available to players. We used Foldit's localization features to create a professional scientist "translation" to use more familiar terminology. A comparison of visualization in Foldit and Foldit Standalone is given in Figure 1. Additionally, Foldit Standalone includes direct access to more advanced Rosetta protein modeling features, adding much-desired functionality that was not possible with other scientific software.

The **Eyewire Tracer Interface** provides users with a familiar split-screen interface for 2D-3D reconstruction, similar to Omni. Eyewire utilizes existing commands and view controls. New user interface elements are introduced in keeping with the Eyewire style guide respecting persistent locations and functions.

5.2 Interoperability

Although games may have their own internal data formats, software support for importing and exporting standard formats is needed. This allows professional scientists to integrate the game-based technology with their workflows and supplement existing software. We have noticed that ease of import and export of data can impact adoption.

Foldit Standalone interoperates with the standard PDB and FASTA formats (along with various other Rosetta-supported data formats). However, an area of remaining work is the inability to take any standard file from the Protein Data Bank and load it into Foldit Standalone for viewing without the user having expert level knowledge of preparing files for use in Rosetta.

The **Eyewire Tracer Interface** works with segmented serial electron microscopy data in standard formats.

5.3 Reusability

Avoiding code duplication between different software versions is important for preventing maintenance issues and reducing the potential for supporting two completely separate programs. Both projects found it useful to structure the citizen scientist and professional scientist functionality so that one is mostly a subset of the other. However, the Foldit and Eyewire accomplished this in opposite ways. While both projects added and removed features, Foldit's main approach was to remove features to make Foldit Standalone, while Eyewire's main approach was to add features to make the Eyewire Tracer Interface. In both cases, code is structured for reuse rather than duplication.

Foldit Standalone is structured as a separate program from Foldit, and mostly as a subset of it, with game elements added on top. The two projects share and reuse most of the Foldit Standalone code. Initially we started with the game and took several approaches to deriving Foldit Standalone from it. Early architectures were more expedient to implement but, in retrospect, turned out to be difficult to maintain. Architecturally, the code for the projects is organized so that Foldit inherits from Foldit Standalone. Thus, all of the code for Foldit Standalone can now exist in one library, and the extensions provided by the game in another. This improves

maintainability, as there is less code duplication, bug fixes made to protein manipulation tools in Foldit Standalone also appear in Foldit, and changes made to game-specific features (such as leaderboards or networking) will not introduce unexpected behavior in Foldit Standalone.

The **Eyewire Tracer Interface**, in contrast, is part of one web application separated into different platforms by user permissions. The game Eyewire is mostly a subset of the Eyewire Tracer Interface, which adds functionality for professional scientists.

5.4 Scalability

It is essential for software to support the large problem sizes and data sets that professional scientists work on.

Foldit Standalone uses multiple threads and animations to keep the user interface responsive while protein computations are carried out, hiding delays as protein size scales. However, there is a performance impact when proteins greater than three hundred residues are loaded, limiting the user's ability to smoothly interact with the molecule of interest. These limitations result from the integrated nature of the Foldit molecular viewing interface with the back-end Rosetta modeling force-field used to energetically evaluate the molecule. Foldit is continually exploring means of manipulating ever larger protein structures at interactive rates, including simplifying geometry and trimming regions of larger problems, though more work remains to be done.

The **Eyewire Tracer Interface** has benefited from Eyewire's substantial optimization to scale to larger neurons, as well as developed tools that facilitate coordinated activity at the whole cell overview level. One major challenge in Eyewire's development was the sheer size of each task. Delivering each of the three 2D planes of a cube took 10-11 MB, or up to 30 MB per cube. Client-side meshing code was written to avoid downloading (sometimes large) meshes, and new ways to compress the data were devised. In all, data delivery shrunk from a 30+ MB maximum to about 5 MB maximum per task. This new "hypersquare" image delivery system enabled a new project that required the octupling of displayed image size in a new dataset based on zebrafish. A new segmentation tool called "Msty", behaving as an automated player, was also added.

5.5 Separability

In terms of game systems, we have noticed that having paid or administrative project members competing directly against players can impact perception of game fairness. Support was needed to prevent citizen and professional scientists from competing in-game.

Foldit Standalone is used offline, disconnected from the Foldit website and game competition and rewards.

The **Eyewire Tracer Interface** is accessed through the standard Eyewire website, although Tracers do not appear on the Eyewire leaderboard. However, Tracers do win badges; according to discussions with Tracers, they are indifferent to points, yet find badges "entertaining" and good for a break between long sets of cubes.

6 CONCLUSION

In this work we have identified an emerging *use of game-based technology*, in the form of *citizen science games* being repurposed for use by professional scientists, and offered evidence that this is

useful to professional scientists. We have noted several key design principles based on our experience repurposing two games.

It is possible that other existing citizen science games may be beneficially used by professional scientists, if they are not already—for example, using Phylo [23] for sequence alignment or EteRNA [29] for RNA design. As citizen science projects are incorporated into AAA games (such as EVE Online's Project Discovery [6]), scientific software may become even more engaging. Where citizen science has commonly involved assisting professional scientists in data gathering, data analysis, and problem solving, in this work we consider that the software artifacts developed for citizen science projects have value to professional scientists as well. Thus, there may be other yet unforeseen ways in which citizen science can benefit professional scientists, or benefit other domains such as science education.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under grant no. 1629879. This work was supported by National Institutes of Health grants 1UH2CA203780 and 1UH2CA203710, IARPA D16PC00005, and RosettaCommons.

REFERENCES

- [1] Paul D. Adams, Pavel V. Afonine, Gábor Bunkóczi, Vincent B. Chen, Ian W. Davis, Nathaniel Echols, Jeffrey J. Headd, Li-Wei Hung, Gary J. Kapral, Ralf W. Grosse-Kunstleve, Airlie J. McCoy, Nigel W. Moriarty, Robert Oeffner, Randy J. Read, David C. Richardson, Jane S. Richardson, Thomas C. Terwilliger, and Peter H. Zwart. 2010. PHENIX: a comprehensive Python-based system for macromolecular structure solution. *Acta Crystallographica Section D: Biological Crystallography* 66, 2 (Feb. 2010), 213–221.
- [2] Manda Banerji, Ofer Lahav, Chris J. Lintott, Filipe B. Abdalla, Kevin Schawinski, Steven P. Bamford, Dan Andreescu, Phil Murray, M. Jordan Raddick, Anze Slosar, Alex Szalay, Daniel Thomas, and Jan Vandenberg. 2010. Galaxy Zoo: reproducing galaxy morphologies via machine learning. *Monthly Notices of the Royal Astronomical Society* 406, 1 (2010), 342–353.
- [3] Jonathan Barone, Colin Bayer, Rowan Copley, Nova Barlow, Matthew Burns, Sundipta Rao, Georg Seelig, Zoran Popović, Seth Cooper, and Nanocrafter Players. 2015. Nanocrafter: design and evaluation of a DNA nanotechnology game. In *Proceedings of the 10th International Conference on the Foundations of Digital Games*.
- [4] Amaury Belin and Yannick Prié. 2012. DIAM: towards a model for describing appropriation processes through the evolution of digital artifacts. In *Proceedings of the Designing Interactive Systems Conference (DIS '12)*. ACM, New York, NY, USA, 645–654.
- [5] Dylan Alexander Carlin, Ryan W. Caster, Xiaokang Wang, Stephanie A. Betzenderfer, Claire X. Chen, Veasna M. Duong, Carolina V. Ryklansky, Alp Alpekin, Nathan Beaumont, Harshul Kapoor, Nicole Kim, Hosna Mohabbot, Boyu Pang, Rachel Teel, Lillian Whithaus, Ilias Tagkopoulos, and Justin B. Siegel. 2016. Kinetic characterization of 100 glycoside hydrolase mutants enables the discovery of structural features correlated with kinetic constants. *PLOS ONE* 11, 1 (Jan. 2016), e0147596.
- [6] CCP Games. 2003. EVE Online. (2003).
- [7] Dennis Chao. 2001. Doom as an interface for process management. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '01)*. ACM, New York, NY, USA, 152–157.
- [8] 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 (Aug. 2010), 756–760.
- [9] Seth Cooper, Adrien Treuille, Janos Barbero, Andrew Leaver-Fay, Kathleen Tuite, Firas Khatib, Alex Cho Snyder, Michael Beenen, David Salesin, David Baker, Zoran Popović, and Foldit Players. 2010. The challenge of designing scientific discovery games. In *Proceedings of the 5th International Conference on the Foundations of Digital Games (FDG '10)*. ACM, Monterey, California, 40–47.
- [10] Silvia Crivelli, Oliver Kreylos, Bernd Hamann, Nelson Max, and Wes Bethel. 2004. ProteinShop: a tool for interactive protein manipulation and steering. *Journal of Computer-Aided Molecular Design* 18, 4 (2004), 271–285.
- [11] W. L. DeLano. 2002. The PyMOL molecular graphics system. (2002).

- [12] Alan Dix. 2007. Designing for appropriation. In *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCL...But Not As We Know It - Volume 2 (BCS-HCI '07)*. BCS Learning & Development Ltd., Swindon, UK, 27–30.
- [13] Christopher B. Eiben, Justin B. Siegel, Jacob B. Bale, Seth Cooper, Firas Khatib, Betty W. Shen, Foldit Players, Barry L. Stoddard, Zoran Popović, and David Baker. 2012. Increased Diels-Alderase activity through backbone remodeling guided by Foldit players. *Nature Biotechnology* 30, 2 (2012), 190–192.
- [14] Paul Emsley and Kevin Cowtan. 2004. Coot: model-building tools for molecular graphics. *Acta Crystallographica Section D: Biological Crystallography* 60, 12 (2004), 2126–2132.
- [15] Chiara Franzoni and Henry Sauermann. 2014. Crowd science: the organization of scientific research in open collaborative projects. *Research Policy* 43, 1 (Feb. 2014), 1–20.
- [16] Mirosław Gilski, Maciej Kazmierczyk, Szymon Krzywda, Helena Ząbranská, Seth Cooper, Zoran Popović, Firas Khatib, Frank DiMaio, James Thompson, David Baker, Iva Pichová, and Mariusz Jaskolski. 2011. High-resolution structure of a retroviral protease folded as a monomer. *Acta Crystallographica Section D: Biological Crystallography* D67, 11 (Nov. 2011), 907–914.
- [17] GlassLab. 2013. SimCityEDU. (2013).
- [18] Benjamin M. Good and Andrew I. Su. 2011. Games with a scientific purpose. *Genome Biology* 12, 12 (2011), 135.
- [19] Sydney R. Gordon, Elizabeth J. Stanley, Sarah Wolf, Angus Toland, Sean J. Wu, Daniel Hadidi, Jeremy H. Mills, David Baker, Ingrid Swanson Pultz, and Justin B. Siegel. 2012. Computational design of an α -gladin peptidase. *Journal of the American Chemical Society* 134, 50 (Dec. 2012), 20513–20520.
- [20] Moritz Helmstaedter, Kevin L. Briggman, and Winfried Denk. 2011. High-accuracy neurite reconstruction for high-throughput neuroanatomy. *Nature Neuroscience* 14, 8 (Aug. 2011), 1081–1088.
- [21] Scott Horowitz, Brian Koepnick, Raoul Martin, Agnes Tymieniecki, Amanda A. Winburn, Seth Cooper, Jeff Flatten, David S. Rogawski, Nicole M. Koropatkin, Tsinaikab T. Hailu, Neha Jain, Philipp Koldewey, Logan S. Ahlstrom, Matthew R. Chapman, Andrew P. Sikkema, Meredith A. Skiba, Finn P. Maloney, Felix R. M. Beinlich, Foldit Players, University of Michigan students, Zoran Popović, David Baker, Firas Khatib, and James C. A. Bardwell. 2016. Determining crystal structures through crowdsourcing and coursework. *Nature Communications* 7 (2016), 12549.
- [22] Narayanan Kasthuri, Kenneth Jeffrey Hayworth, Daniel Raimund Berger, Richard Lee Schalek, José Angel Conchello, Seymour Knowles-Barley, Dongil Lee, Amelio Vázquez-Reina, Verena Kaynig, Thouis Raymond Jones, Mike Roberts, Josh Lyskowski Morgan, Juan Carlos Tapia, H. Sebastian Seung, William Gray Roncal, Joshua Tzvi Vogelstein, Randal Burns, Daniel Lewis Sussman, Carey Eldin Priebe, Hanspeter Pfister, and Jeff William Lichtman. 2015. Saturated reconstruction of a volume of neocortex. *Cell* 162, 3 (July 2015), 648–661.
- [23] Alexander Kawrykow, Gary Roumanis, Alfred Kam, Daniel Kwak, Clarence Leung, Chu Wu, Eleyine Zarour, Luis Sarmenta, Mathieu Blanchette, Jérôme Waldspühl, and Phyllo Players. 2012. Phyllo: a citizen science approach for improving multiple sequence alignment. *PLoS ONE* 7, 3 (March 2012), e31362.
- [24] Firas Khatib, Frank DiMaio, Foldit Contenders Group, Foldit Void Crushers Group, Seth Cooper, Maciej Kazmierczyk, Mirosław Gilski, Szymon Krzywda, Helena Ząbranská, Iva Pichová, Gray Thompson, Zoran Popović, Mariusz Jaskolski, and David Baker. 2011. Crystal structure of monomeric retroviral protease solved by protein folding game players. *Nature Structural and Molecular Biology* 18, 10 (Sept. 2011), 1175–1177.
- [25] Jinseop S. Kim, Matthew J. Greene, Aleksandar Zlateski, Kisuk Lee, Mark Richardson, Srinivas C. Turaga, Michael Purcaro, Matthew Balkam, Amy Robinson, Bardia F. Behabadi, Michael Campos, Winfried Denk, H. Sebastian Seung, and EyeWriters. 2014. Space-time wiring specificity supports direction selectivity in the retina. *Nature* 509, 7500 (May 2014), 331–336.
- [26] Robert Kleffner, Jeff Flatten, Andrew Leaver-Fay, David Baker, Justin B. Siegel, Firas Khatib, and Seth Cooper. 2017. Foldit Standalone: a video game-derived protein structure manipulation interface using Rosetta. *Bioinformatics* (2017).
- [27] Belinda Lange, Sheryl Flynn, and Albert Rizzo. 2009. Initial usability assessment of off-the-shelf video game consoles for clinical game-based motor rehabilitation. *Physical Therapy Reviews* 14, 5 (Oct. 2009), 355–363.
- [28] Andrew Leaver-Fay, Michael Tyka, Steven M. Lewis, Oliver F. Lange, James Thompson, Ron Jacak, Kristian Kaufman, P. Douglas Renfrew, Colin A. Smith, Will Sheffler, Ian W. Davis, Seth Cooper, Adrien Treuille, Daniel J. Mandell, Florian Richter, Yih-En Andrew Ban, Sarel J. Fleishman, Jacob E. Corn, David E. Kim, Sergey Lyskov, Monica Berrondo, Stuart Mentzer, Zoran Popović, James J. Havranek, John Karanicolas, Rhiju Das, Jens Meiler, Tanja Kortemme, Jeffrey J. Gray, Brian Kuhlman, David Baker, and Philip Bradley. 2011. ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules. *Methods in Enzymology* 487 (2011), 545–574.
- [29] Jeehyung Lee, Wipapat Kladwang, Minjae Lee, Daniel Cantu, Martin Azizyan, Hanjoo Kim, Alex Limpaecher, Sungroh Yoon, Adrien Treuille, Rhiju Das, and EteRNA Participants. 2014. RNA design rules from a massive open laboratory. *Proceedings of the National Academy of Sciences* 111, 6 (Feb. 2014), 2122–2127.
- [30] Edgar Luttmann, Daniel L. Ensign, Vishal Vaidyanathan, Mike Houston, Noam Rimon, Jeppe Øland, Guha Jayachandran, Mark Friedrichs, and Vijay S. Pande. 2009. Accelerating molecular dynamic simulation on the cell processor and Playstation 3. *Journal of Computational Chemistry* 30, 2 (Jan. 2009), 268–274.
- [31] Mojang. 2016. Minecraft: Education Edition. (2016).
- [32] Antônio Carlos A. Mól, Carlos Alexandre F. Jorge, and Pedro M. Couto. 2008. Using a game engine for VR simulations in evacuation planning. *IEEE Computer Graphics and Applications* 28, 3 (May 2008), 6–12.
- [33] Eric F. Pettersen, Thomas D. Goddard, Conrad C. Huang, Gregory S. Couch, Daniel M. Greenblatt, Elaine C. Meng, and Thomas E. Ferrin. 2004. UCSF Chimera—a visualization system for exploratory research and analysis. *Journal of Computational Chemistry* 25, 13 (Oct. 2004), 1605–1612.
- [34] 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).
- [35] Ghanasyam Rallapalli, Fraxinus Players, Diane Go Saunders, Kentaro Yoshida, Anne Edwards, Carlos A. Lugo, Steve Collin, Bernardo Clavijo, Manuel Corpas, David Swarbreck, Matthew Clark, J. Allan Downie, Sophien Kamoun, Team Cooper, and Dan MacLean. 2015. Lessons from Fraxinus, a crowd-sourced citizen science game in genomics. *eLife* 4 (2015), e07460.
- [36] Ingmar H. Riedel-Kruse, Alice M. Chung, Burak Dura, Andrea L. Hamilton, and Byung C. Lee. 2011. Design, engineering and utility of biotic games. *Lab on a Chip* 11, 1 (2011), 14–22.
- [37] Carol A. Rohl, Charlie E. M. Strauss, Kira M. S. Misura, and David Baker. 2004. Protein structure prediction using Rosetta. *Methods in Enzymology* 383 (2004), 66–93.
- [38] Rachel Welles Shearer. 2009. *OMNI: visualizing and editing large-scale volume segmentations of neuronal tissue*. Master's. MIT.
- [39] Akihiko Shirai, Erik Geslin, and Simon Richir. 2007. WiiMedia: motion analysis methods and applications using a consumer video game controller. In *Proceedings of the 2007 ACM SIGGRAPH Symposium on Video Games (Sandbox '07)*. ACM, New York, NY, USA, 133–140.
- [40] Mohd Fairuz Shiratuddin and Walid Thabet. 2002. Virtual office walkthrough using a 3D game engine. *International Journal of Design Computing* 4 (2002).
- [41] Justin B. Siegel, Amanda Lee Smith, Sean Poust, Adam J. Wargacki, Arren Bar-Even, Catherine Louw, Betty W. Shen, Christopher B. Eiben, Huu M. Tran, Elad Noor, Jasmine L. Gallaher, Jacob Bale, Yasuo Yoshikuni, Michael H. Gelb, Jay D. Keasling, Barry L. Stoddard, Mary E. Lidstrom, and David Baker. 2015. Computational protein design enables a novel one-carbon assimilation pathway. *Proceedings of the National Academy of Sciences* 112, 12 (March 2015), 3704–3709.
- [42] Brian L. Sullivan, Christopher L. Wood, Marshall J. Iliff, Rick E. Bonney, Daniel Fink, and Steve Kelling. 2009. eBird: a citizen-based bird observation network in the biological sciences. *Biological Conservation* 142, 10 (Oct. 2009), 2282–2292.
- [43] Mark C. Surles, Jane S. Richardson, David C. Richardson, and Fred P. Brooks. 1994. Sculpting proteins interactively: continual energy minimization embedded in a graphical modeling system. *Protein Science* 3 (1994), 198–210.
- [44] Valve Corporation. 2011. Learn With Portals. (2011).
- [45] Ashwin Vishwanathan, Kayvon Daie, Alexandro D. Ramirez, Jeff W. Lichtman, Emre R. F. Aksay, and H. Sebastian Seung. 2017. Electron microscopic reconstruction of functionally identified cells in a neural integrator. *Current Biology* 27, 14 (July 2017), 2137–2147.e3.
- [46] 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.
- [47] Jijun Wang, Michael Lewis, and Jeffrey Gennari. 2003. Emerging areas: urban operations and UCAVs: a game engine based simulation of the NIST urban search and rescue arenas. In *Proceedings of the 35th Conference on Winter Simulation: Driving Innovation (WSC '03)*. Winter Simulation Conference, New Orleans, Louisiana, 1039–1045.
- [48] Andrew J. Westphal, Anna L. Butterworth, Christopher J. Snead, Nahide Craig, David Anderson, Steven M. Jones, Donald E. Brownlee, Richard Farnsworth, and Michael E. Zolensky. 2005. Stardust@home: a massively distributed public search for interstellar dust in the stardust interstellar dust collector.
- [49] Sean J. Wu, Christopher B. Eiben, John H. Carra, Ivan Huang, David Zong, Peixian Liu, Cindy T. Wu, Jeff Nivala, Josef Dunbar, Tomas Huber, Jeffrey Senft, Rowena Schokman, Matthew D. Smith, Jeremy H. Mills, Arthur M. Friedlander, David Baker, and Justin B. Siegel. 2011. Improvement of a potential anthrax therapeutic by computational protein design. *Journal of Biological Chemistry* 286, 37 (Sept. 2011), 32586–32592.