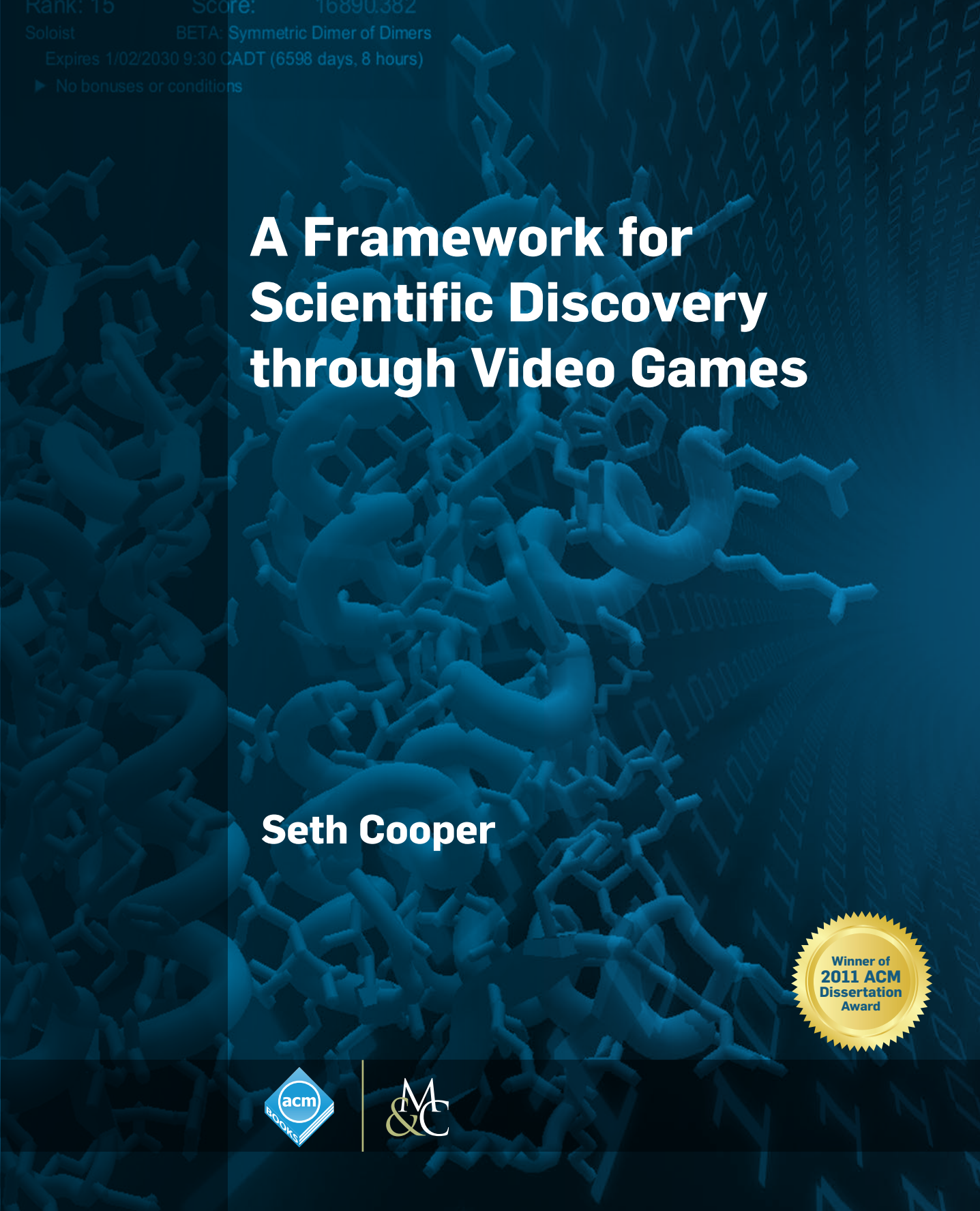

A Framework for Scientific Discovery through Video Games

Seth Cooper

Winner of 2011 ACM Dissertation Award

acm BOOKS | &MC

A Framework for Scientific Discovery through Video Games

ACM Books

Editor in Chief

M. Tamer Özsu, *University of Waterloo*

ACM Books is a new series of high-quality books for the computer science community, published by ACM in collaboration with Morgan & Claypool Publishers. ACM Books publications are widely distributed in both print and digital formats through booksellers and to libraries (and library consortia) and individual ACM members via the ACM Digital Library platform.

This book by Seth Cooper is a revised version of the dissertation that won the 2011 ACM Doctoral Dissertation Award. Other books in the series include those listed below.

A Framework for Scientific Discovery through Video Games

Seth Cooper, *University of Washington*
2014

Trust Extension as a Mechanism for Secure Code Execution on Commodity Computers

Bryan Jeffrey Parno, *Microsoft Research*
2014

Embracing Interference in Wireless Systems

Shyamnath Gollakota, *University of Washington*
2014

A Framework for Scientific Discovery through Video Games

Seth Cooper
University of Washington

ACM Books #3

A Framework for Scientific Discovery through Video Games

Seth Cooper

books.acm.org
www.morganclaypool.com

ISBN: 978-1-62705-504-8 print
ISBN: 978-1-62705-505-5 ebook
ISBN: 978-1-62705-506-2 ePub
Series ISSN: (to come)

DOIs: 10.1145/2625848 Book
10.1145/2625848.2625849 Preface
10.1145/2625848.2625850 Chapter 1
10.1145/2625848.2625851 Chapter 2
10.1145/2625848.2625852 Chapter 3
10.1145/2625848.2625853 Chapter 4
10.1145/2625848.2625854 Chapter 5
10.1145/2625848.2625855 Chapter 6
10.1145/2625848.2625856 Chapter 7
10.1145/2625848.2625857 Bibliography

A publication in the ACM Books series, #3
Editor in Chief: M. Tamer Özsu, *University of Waterloo*

First Edition

10 9 8 7 6 5 4 3 2 1

To my parents, Harris and Beth Cooper

Contents

Preface

When we first set out to create Foldit over six years ago, it wasn't clear that a game-based approach to scientific discovery would work. So we planned from the start for the game to be continually adapting and changing, in order to keep improving based on the lessons we'd learn. It took several years of design, development, and continued iteration from a team of computer scientists and biochemists until the game was at a point where we made our first exciting discovery. The nature of the challenging problems we were facing required this time and refinement to solve.

We are now seeing a number of other games that allow players to contribute to scientific research. Much of this growth has been in fields related to biology and biochemistry: EteRNA for designing RNA shapes, EyeWire for mapping neurons, and Phylo for aligning genetic sequences. Each of these games has had exciting scientific results produced by gameplay. Games are being applied in other areas as well, such as in the Algoraph suite of games for solving graph theory problems. I have been involved in the development of two more science games: Nanocrafter, which aims to push the frontiers of DNA-based synthetic biology, and Flow Jam, which allows players to help formally verify software.

The Foldit community has continued to grow and adapt to the new scientific challenges that we have posed. We have continued to look further into the design of synthetic proteins and their applications for health and understanding proteins. The Foldit team is continually grateful to the player community for their creativity, problem solving, and enthusiasm. Their brainpower and fresh insights have been critical to the successes of the project; without the players, there would be no game.

Scientific discovery games are an exciting part of the growing effort towards engaging the public in science. I look forward to new ways for anyone with an interest and a passion to contribute to scientific discovery.

Acknowledgments

There are many people to thank for helping and supporting me during the creation of this book.

- my advisor, Zoran Popović, for his guidance, support, and the opportunity to work on such a great video game as part of my research;
- David Baker for his expertise and time for helping make this a successful project;
- my committee members, David Salesin, John Zahorjan, and Ethan Merrit, for their valuable feedback throughout the project;
- the Foldit development team—Adrien Treuille and Janos Barbero were instrumental at the very beginning of the project, starting off on the considerable task of creating Foldit;
- the members of the Baker lab for lending their considerable biochemical expertise to the project. Firas Khatib contributed excellent biochemical insight and interaction with the Foldit community, and Andrew Leaver-Fay was always available and helpful in understanding the Rosetta codebase;
- the community of Foldit players for their enormous contribution to this work, for always surprising me with their ingenuity, and for their patience with the development of the game; and
- my parents, Harris and Beth Cooper, my sister, Emily Cooper, and my wife, Fayette Shaw, for all their help and support throughout school.

My work was supported by the National Science Foundation, DARPA, the Howard Hughes Medical Institute, NVIDIA, Electronic Arts, Sony, Microsoft, and Adobe.

Seth Cooper
July 2014

1 Introduction

1.1 Motivation

Despite the massive amounts of computational power available, many difficult scientific problems still remain computationally intractable. Fortunately, people possess great skill in problem solving and creativity, and individual problem solving skills can be augmented by working together in groups. However, only a small population of people are involved in scientific inquiry and advancing science.

The following questions arise: (1) How much more scientific advancement would be possible if more people were involved? (2) Can we integrate what people and computers, respectively, do well? We would like to maximize the effectiveness of this human-computer symbiosis, to find places were computational power is most useful and where human ability can best be applied. People and computers are often good at solving different types of problems; for example, a person would likely translate a passage of text more naturally, and a computer would likely be able to numerically optimize a function faster. We would like to be able to combine the best abilities of each in order to solve challenging problems that neither could alone.

The goal of this book is to determine if it is possible to design the coevolution of human-computer symbiosis to solve currently open problems in science. Two particular areas where humans can excel are spatial reasoning and creativity. People are able to reason spatially by forming mental models of objects, their environment, and the spatial relationships between them [Byrne and Johnson-Laird 1989, Tversky 1993].

People enjoy expressing their creativity, and many successful video games require players to think about objects in space and their spatial relationships to each other. Tetris[1] is a popular example of this. There are many physical puzzles that rely on spatial reasoning as well, such as Rubik's Cube[2] and many sliding block puzzles. Recently, there has been a rise in popularity of video games whose explicit purpose is to help

1. http://www.tetris.com/; last retrieved May 2014.

2. http://www.rubiks.com/; last retrieved May 2014.

people train their cognitive skills, such as spatial reasoning. Brain Age[3] and Big Brain Academy[4] are examples of this.

A scientific field that naturally requires spatial reasoning and creativity for problem solving is biochemistry. Many problems in biochemistry are fundamentally spatial structural problems, particularly when dealing with protein structures. Proteins are important to biochemistry and our understanding of life itself, because they are indispensable to living systems and perform many important tasks in the cell, including structural, transport, defensive, and catalytic roles. The way proteins achieve their function is due to their shapes and how they interact with other molecules. They involve the relationships between physical objects in three-dimensional space; a protein's structure determines its function [Zhang and Kim 2003].

To explore the potential of this human-computer framework for solving scientific problems, we have developed Foldit,[5] an online video game that casts protein structure manipulation as a puzzle solving competition. The game tries to predict naturally occurring protein structures and to design novel proteins not previously seen in nature. In order to achieve this goal, the game gives players the ability to manipulate and optimize protein structures while competing and collaborating with other players to discover the best structures. Foldit's YouTube channel can be found at http://www.youtube.com/user/uwfoldit; http://www.youtube.com/watch?v=lGYJyur4FUA gives a good introduction to the game.

1.2 Problem Statement

1.2.1 Game Design Problem

Designing a game for scientific discovery presents many distinct challenges. One of the primary purposes of using a game is to maximize the engagement and retention of the players. However, it is not enough to simply make the game as fun as possible; this goal must also be balanced with the need for relevant scientific outcomes. For most games, the designer is free to make decisions based only on what will make the game fun. In a scientific discovery game, the tension between the freedom to design for engagement and the realism of the scientific constraints is a key challenge. Thus, an important question is, how can we design a game that is both engaging and produces useful results? We would also like to know what kinds of problems would lend themselves

3. http://brainage.com/; last retrieved May 2014.

4. http://bigbrainacademy.com/; last retrieved May 2014.

5. http://fold.it/

to such contributions by non-scientists, and how can we identify these problems and map them onto a game.

We presume that the game players begin without any knowledge of the scientific field the game is based in. Given this, we would like for the players to gain the domain knowledge necessary to make a contribution to a challenging scientific problem quickly. This is not necessarily general expertise in the subject area or formal scientific expertise. Players may develop their own specialized form of expertise unique to the problem presentation within the game. How can the game best support the training of players to the point where they can make a contribution, and integrate players into the scientific process? We would like to use structures from games to teach players, and keep players interested and involved long-term. We would also like to spread the expertise gained by experienced players and to help new players learn from it.

The game's architecture should support the coevolution of both the players and the game itself. In this way, as the players adapt to the game by gaining experience in how the game works and solving the problems presented, the game can also adapt to how the players best use it to become a better tool. How can we allow for this coevolution of the game and the player base?

1.2.2 Biochemistry Discovery Problem

Predicting protein structures computationally is a central goal for computational biochemists because so much can be understood about a protein's function once its structure is known, and because it is so challenging to observe a protein's structure directly. Proteins—chains of smaller molecules called amino acids—are central to biochemistry because they are the primary chemical for almost all cellular processes. Understanding a protein's structure is necessary to understand its functions, because a protein's shape determines how it will interact with other molecules. Thus, an important problem in biochemistry is the *protein structure prediction* problem: given the sequence of amimo acids that make up a protein, what is its structure? It is possible to experimentally determine a protein's structure through methods such as X-Ray Crystallography and Nuclear Magnetic Resonance spectroscopy. Experimental methods, however, can be costly, time consuming, and difficult. This makes computational methods that can accurately predict a protein's structure an attractive solution. However, computational methods are often intractable; the vast number of possible shapes a protein can take make it difficult to find the correct structure. The spatial nature of this problem makes it a good candidate for the application of human spatial reasoning.

A related problem is that of protein *design*: given a desired function for a protein, what is the amino acid sequence that, when folded, will carry it out? In this case, computational methods are even more attractive. Synthesizing proteins to test every

design would be prohibitively expensive, while computational methods can allow us to filter out designs that are not likely to work. Protein design has implications for drug design, in inhibitors and vaccines, for biofuel design, in enzymes, and for other areas. Human creativity can be applied to help create novel proteins that did not exist before.

1.3 Outline

In this book we will show the effectiveness of the game-based framework as an approach to scientific discovery. Chapter 2 discusses the literature related to this book. Chapter 3 gives an overview of the game-based framework used in this research, describing the dual goals of engagement and scientific relevance, and the coevolution approach we take. A discussion of using this framework for problem solving as applied to protein structure prediction is given in Chapter 4, and we show that players can predict the unknown structures of naturally occurring proteins, even where all previous methods have failed. Further discussion of applying this framework to leverage player creativity for protein design is given in Chapter 5, and we show that players can become an integral part of the design of novel and effective proteins. Chapter 6 describes an approach to allowing players to codify and automate their strategies, and we show that players can socially develop highly effective algorithms. Finally, Chapter 7 provides a summary and discusses possible future directions for research.

2 Related Literature

This book is related to several bodies of existing literature, including volunteer computing, human computation, serious games, computational biochemistry, and visualization and interaction.

2.1 Volunteer Computing and Human Computation

Volunteer computing is a method by which volunteers are able to donate their computer's spare time and space to various projects. The volunteer computing model has risen in popularity recently, and has allowed scientists access to unprecedented amounts of computational power. One of the oldest and largest volunteer computing projects is SETI@home[1] [Sullivan III et al. 1997]. This project uses a screensaver to analyze radio telescope data. There is an open source platform for developing volunteer computing projects, the Berkeley Open Infrastructure for Network Computing (BOINC),[2] which allows users to manage and share their computer's resources between the many projects using the platform [Anderson 2004]. BOINC projects have a variety of goals, from climate prediction [Stainforth et al. 2005] to searching for pulsars [Knispel et al. 2010].

By using the volunteer computing model, projects not only gain access to massive computation, but also allows the public to make contributions to science. However, with this model, their contributions are mostly passive—they don't even have to be at their computer. This work aims to use not only the power of networks of computers, but also that of networks of humans, and allow people to make active contributions to science.

There has been work recently on leveraging a human workforce for computational tasks that computers are not yet able to perform satisfactorily. A more general field

1. http://setiathome.ssl.berkeley.edu/; last retrieved May 2014.
2. http://boinc.berkeley.edu/; last retrieved May 2014.

of "human computation" or "distributed thinking" is emerging. On a smaller scale, augmenting automated heuristics with interactive human input can help to solve basic spatial problems [Anderson et al. 2000, Lesh et al. 2005]. On a larger scale, general tasks desired for humans to perform are posted online, and users can determine which tasks they would like to perform. Amazon's Mechanical Turk[3] is one example of such a system, where users are actually paid to perform tasks. Example tasks include translation of text, rating search results, and determining the tone of an article. Bossa is an open source system for managing similar user tasks.[4]

In this context, there has been much interest in using games as a means of motivating people to perform tasks that are currently difficult for computers. One particularly active area is in computer vision and image recognition. Humans are particularly adept at reading words in images and determining the objects in a scene, when compared with current computational methods. The difference in ability is strong enough that vision-based tests are often used as a proof of humanity with CAPTCHAs [Ahn et al. 2003].

Games such as the ESP game [von Ahn and Dabbish 2004], Peekaboom [von Ahn et al. 2006], and Google Image Labeler[5] use human image-recognition ability to produce labeled images from gameplay. Image recognition has also been used for finding particular features of interest in scientific data, such as looking for signs of interstellar dust [Westphal et al. 2010], measuring and aligning features on a planet's surface,[6] and classifying galaxy shapes.[7] These projects have been successful in motivating players to sift through large image sets, which would otherwise be a mundane task.

Some games have approached other types of problems. Pebble It[8] is a game which studies human solutions to the graph pebbling problem, with the goal of developing better algorithms to solve it [Cusack et al. 2006]. Outside of games, some work has examined how to fit human problem solving into various optimization problems [Anderson et al. 2000, Lesh et al. 2005]. This work is different because it leverages a deeper human problem solving ability to create interesting scientific results.

3. http://www.mturk.com/; last retrieved May 2014.

4. http://boinc.berkeley.edu/trac/wiki/BossaIntro; last retrieved May 2014.

5. http://images.google.com/imagelabeler/; last retrieved May 2014.

6. NASA Be A Martian, http://beamartian.jpl.nasa.gov/; last retrieved May 2014.

7. Galaxy Zoo, http://www.galaxyzoo.org/; last retrieved May 2014.

8. http://pebbleit.hope.edu/; last retrieved May 2014.

2.2 Serious Games and Gamification

Recently, a field known as "serious games" has been identified. The most general definition is any game that has a purpose beyond simply entertaining the player; however, it often connotes games whose purpose is training or education. The line between game and simulation or application is also not always well defined. A game-based approach is appealing because games are meant to be engaging and motivating. Furthermore, other fields are taking advantage of the fact that gaming has pushed the limits of interactive simulation and authoring technology, such as 3D engines [Susi et al. 2007]. Some of the wide-ranging applications of serious games are firefighter training [Backlund et al. 2007], raising awareness of social issues,[9] and military recruitment.[10] In this book, the main goal is to generate useful scientific discoveries; however, other aspects of game design, such as the requirement that the game be fun, contribute to achieving this goal, as the results rely on players playing the game.

One particular subgenre of serious games is games for health. Playing games has, in some settings, been shown to be beneficial to the player's health. Games have been shown to be useful for rehabilitation, development, and therapy, and even for distracting patients from pain or bad habits [Adriaenssens et al. 1988, Griffiths 2005]. Nintendo's Wii Fit package[11] is intended to help players improve their personal fitness.

Many games emphasize social interactions as well. Massively multiplayer online games (MMOs), like World of Warcraft[12] and Second Life,[13] often host persistent virtual worlds where players can customize avatars, socialize, and work together with other players. Diverse niche MMOs exist, targeting teens or people interested in racing, allowing people with similar interests to interact [Zenke 2008].

Similarly, Alternate Reality Games (ARGs) engage large groups of people to participating in narratives in the real world [Martin et al. 2006]. Often, multiple forms of technology will be used to coordinate the players, who will be working together towards a common goal. I Love Bees, a popular ARG, had players work together to find payphones and answer prerecorded questions [Terdiman 2004].

9. Darfur is Dying, http://www.darfurisdying.com; last retrieved May 2014.

10. America's Army, http://www.americasarmy.com/; last retrieved May 2014.

11. http://www.nintendo.com/wiifit/; last retrieved May 2014.

12. http://us.battle.net/wow/en/; last retrieved May 2014.

13. http://secondlife.com/; last retrieved May 2014.

2.3 Computational Biochemistry

In the field of biochemistry, many computational methods have been used to study protein folding, predict protein structures, and design new proteins. The most closely related to our work is Rosetta. Rosetta combines structural energy minimization with a Monte Carlo search algorithm to predict native protein structures [Rohl et al. 2004]. Foldit uses the Rosetta software for its energy function, as well as many of the algorithms and functionality for energy minimization and protein manipulation. In order to access massive amounts of computation for searching a protein's large structural space, the volunteer computing project Rosetta@home[14] runs Rosetta's algorithms on volunteer's computers.

Another volunteer computing project, Folding@home,[15] aims to simulate the process of protein folding. The Folding@home project has also been ported to the PS3, and thus has access to powerful hardware and gives gamers the opportunity to help science. Folding@home's approach is based on simulating the molecular dynamics of protein folding [Pande et al. 2003]. One difference to note is that Rosetta and Foldit only attempt to determine the final structures, while Folding@home simulates the folding process. This makes Rosetta and Foldit more amenable to problems like protein design, where one is interested in the final folded structure, while Folding@home's approach is more useful for studying topics like protein misfolding.

Many visualizations for biological molecules have been developed to aid biochemists. Popular visualizations include Corey-Pauling-Koltun (CPK) [Corey and Pauling 1953], which displays the chemical properties of individual atoms, and cartoon or ribbon, which shows a more abstract, stylized version of the backbone's secondary structures [Natarajan et al. 2008]. PyMOL is a popular viewer that gives access to many of these visualizations.[16] These visualizations are tuned for scientists, and may not be appropriate for novices.

One possible approach to manipulating three-dimensional objects is to use widgets. Widgets associate behavior with geometry in the scene, so users can interact directly with the environment [Conner et al. 1992]. The Sculpt system allows users to interact with and guide a protein as it folds in the presence of minimization of a physically plausible energy [Surles et al. 1994]. While Sculpt has some features in common with Foldit, it is a single-user application, while Foldit is a multiplayer game, with many new features.

14. http://boinc.bakerlab.org/rosetta/; last retrieved May 2014.

15. http://folding.stanford.edu/; last retrieved May 2014.

16. The PyMOL Molecular Graphics System, http://www.pymol.org/; last retrieved May 2014

3 Framework

3.1 Introduction

This chapter introduces a general framework for *scientific discovery games*. We present guidelines for mapping a scientific problem into a game, and address the often conflicting goals of engagement and scientific relevance. The driving example is *Foldit*, a game for scientific discovery in biochemistry. We describe the architecture of the game. The architecture is flexible and able to coevolve, along with the game's players, to improve as a tool. We discuss the teaching and reward structures in the game, intended to appeal to a wide variety of players, regardless of biochemistry background.

A scientific discovery game translates a class of computationally difficult scientific problems into puzzles, and provides a game-like mechanism for non-scientist players to help solve these problems. Many traditional aspects of game design apply to scientific discovery games, including the design of introductory levels to draw newcomers and explain game mechanics, the use of a client-server architecture for competition and collaboration, and the requirement that the game be fun. However, unlike games whose goal is entertainment or education, scientific discovery games introduce a unique challenge: *enabling non-scientist natural problem solvers to advance a specific scientific domain*. This challenge influences all aspects of the game design. First, visualization and graphics need to promote human ability to see complex solutions and convey accurate scientific information while remaining accessible to beginners. Second, interaction design must optimize for natural interactions suitable for the human exploration process, while still respecting scientific constraints. Finally, the scoring mechanism needs to be informative enough to promote multiple human strategies, while remaining true to the latest models of the underlying scientific phenomenon. Perhaps the most distinguishing feature and the greatest difficulty of design for this type of game is that the solution to the scientific problem, and thus the solution to the corresponding puzzles, is unknown. Since we do not know the solution *a priori*, we cannot design the game with specific solutions in mind.

To explore this space, we focused on human ability to reason about 3D structures and on the biochemistry domain, where many problems tend to be structural. We developed *Foldit*, a biochemical discovery game. In this chapter, we discuss the framework for Foldit's design, with emphasis on the game's initial focus on protein structure

Figure 3.1 **Foldit webpage.** The front page shows recent news about the game, the top players and groups for the current puzzles, and allows the player to log in.

prediction—determining a protein's shape given its sequence of constituent amino acids. Protein structure prediction involves finding favorable interactions that form when the protein's chemical groups come into contact—essentially a 3D jigsaw puzzle. We believe that humans' innate spatial reasoning ability makes it possible for non-scientists to make useful contributions to this problem. We leverage scientists' knowledge to shape the rules of the game, thus enabling a much larger pool of non-scientists to make discoveries within this framework.

The webpage for Foldit is located at http://fold.it. The front page is shown in Figure 3.1. Foldit was publicly released in May 2008. During the first two years following release, we ran roughly 600 structure prediction puzzles and had over 57,000 players from a wide variety of backgrounds participate.

The rest of this chapter describes our experience designing Foldit, with a special emphasis on the unique challenges posed by making biochemistry problems accessible to anyone. The creation of Foldit was a challenging and multidisciplinary project, drawing together computer science, art, game design and biochemistry. Moreover, we did not know ahead of time which parts of the problem players would be best at solving, or which in-game manipulation tools they would use most effectively. The only way to find out was to have people play Foldit. In order to deal with these and other uncertainties, we took an iterative approach both before and after releasing the game to the public. We have continually evolved the gameplay in response to massive gameplay traces, player feedback and scientists' analysis, and continue even now with this iterative process as we add features and expand the set of biochemical problems to which the Foldit community can contribute.

Games are often designed with an iterative approach, which involves designing, testing, and evaluating repeatedly until the player's experience meets some criteria [Fullerton 2008]. For most games, the main criterion for the player's experience is simply to have fun. Player feedback and playtesting are an integral part of the process, and there are a number of methods of gathering and incorporating this information from players [Ambinder 2009]. We have also continued the design process after the game's release, to incorporate data gathered from the players in a continual process of evolutionary redesigning [Kennerly 2003]. Our work differs from the standard iterative approach in that the game design space is constrained to conform with existing physical models, we include the input of scientists in the evaluation of the game, and we include the long-term coevolution of the players and game in the design.

3.2 Biochemistry Background

Here we provide some background on biochemistry and proteins that will be used throughout the rest of this work.

DNA, a cellular chemical perhaps more widely recognized than proteins, derives its entire purpose in encoding protein sequences. Proteins are coded for by DNA, and are created in the cell as a long chain of *amino acids*. A protein's amino acid sequence is known as its *primary structure*. There are twenty different types of amino acids. Regardless of type, some of atoms making up the amino acid will be the same; these are connected together and form the protein's *backbone*. However, the remaining atoms are different for each type; these extend outward from the backbone and are called *sidechains*. The atoms that make up the sidechains divide the amino acids into two main groups: *hydrophobic*, which prefer to be buried on the interior away from water;

and *hydrophilic*, which prefer to be exposed on the exterior near water. These preferences impact how the protein folds. As the amino acids are connected together, the protein begins to fold up; after the amino acids join together, they are often called *residues*. Local characteristics of the fold are referred to as *secondary structure*. These include: *helices*, which are tightly coiled; *sheets*, which are extended straight; and *loops*, which are everything else. The positions of the atoms making up a folded protein is its *tertiary structure*; the tertiary structure taken in nature is a *native structure*. The native structure is one that is lowest in free energy—it has the most favorable set of chemical interactions. It is well known that sequence determines structure [Anfinsen 1973]. In this book, the term sequence will refer to a protein's primary structure, and structure will refer to its tertiary structure, unless otherwise specified.

3.3 Framework Description

3.3.1 Architecture

Here we give an overview of the architecture of Foldit, which can be seen at a high level in Figure 3.2. Foldit uses a client-server architecture. Players must create an account and download the game in order to play. The game then communicates with a central server to send information about the local player and get information about other players.

Scientists post problems to the server; in the case of Foldit, these are protein structures for which the players are meant to find the native structures. An initial protein structure is associated with metadata such as a title and description, and parameterization such as which energy function terms to use. We call these *puzzles*, and they are posted on the server for a fixed amount of time (usually a week). While a puzzle is active, players can download it and interactively reshape the protein to try to achieve the best score. This often requires significant changes to the puzzle structures, which are given in various partially-folded states, and in some cases need to be completely refolded from a straight line. Players' structures, or *solutions*, are reported back to the server, and players are ranked against other players who are playing the same puzzle. Players can form groups with which to share their solutions through the server, allowing them to work together to find even better solutions than they could working alone. When one player *shares* a solution by uploading it to the server, other players in the same group are able to see it and download it. The social aspect of the game is supported by in-game chat, a website with forums, and a player-created wiki. At the close of a puzzle, the solution data is aggregated, and presented to the scientists for analysis.

The game is designed to be flexible, and the client allows automatic updating so that we can continually evolve the gameplay. The puzzle posting cycle and automatic

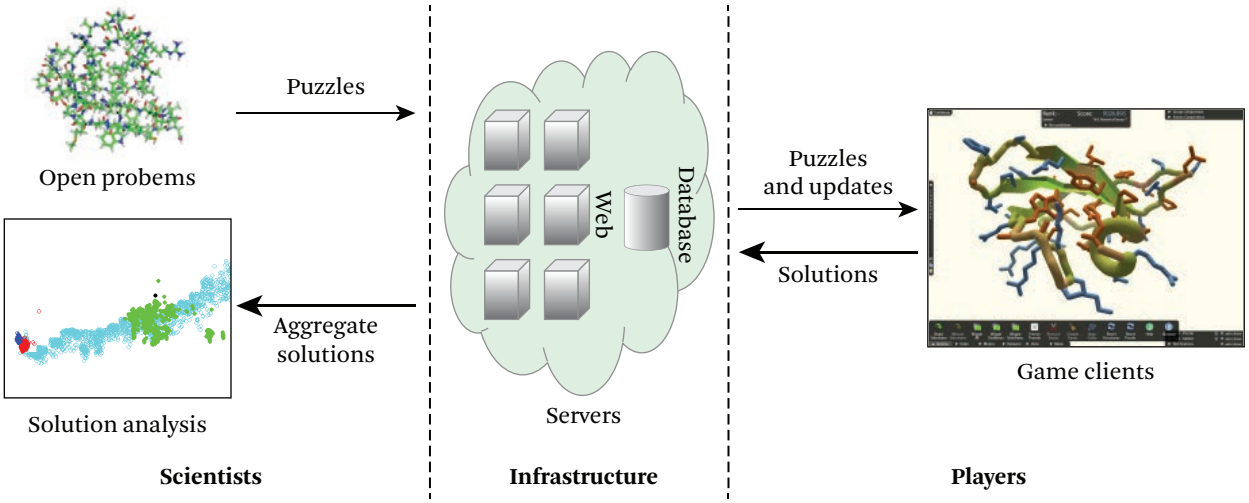

Figure 3.2 **Overview of architecture for scientific discovery games.** The biochemistry team provides structure prediction and design problems for the server. These problems become puzzles and are sent to each player's client. Players collaborate and compete to solve these problems and upload their solutions to the server, where they are aggregated and sent back to the biochemistry team for analysis. This analysis can then be used to improve the design of the game and puzzles. (Figure from Cooper et al. [2010b])

updates allow us to respond to not only player feedback, but also to scientists' analysis, as we introduce and refine gameplay elements.

Foldit is built on top of the Rosetta molecular modeling suite which has proven useful at a wide variety of protein modeling tasks [Rohl et al. 2004, Bradley et al. 2005, Qian et al. 2007, Kuhlman et al. 2003]. The suite contains an energy function which captures the interaction energies between protein elements, as well as a set of structural optimization subroutines. For protein structure prediction, structures closer to the native structure will have a lower energy than structures further away from it. Foldit uses this state-of-the-art energy function to compute player's scores, and also takes advantage of the optimization routines Rosetta makes available.

3.3.2 Coevolution Strategy

In order to arrive at the current state of Foldit, we took an coevolution approach to the game's design. Given the complexity of this undertaking, we realized that it was unlikely that all our initial decisions would be the best. There are three major groups relevant to our approach: (1) the scientists whose problems the game is meant to help solve; (2) the

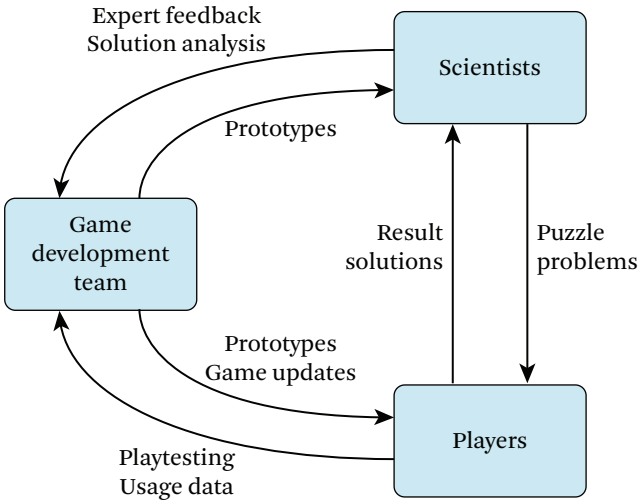

Figure 3.3 **Overview of the interactions between the three iterative design groups.** (Figure from Cooper et al. [2010b])

players; and (3) the game development team. The development team must incorporate feedback from the players to make sure the game is understandable and fun, and from the scientists to make sure that the results produced will be useful to them. An overview of the interactions between these three groups is given in Figure 3.3.

During the game's initial development, the development team and scientists must work together closely to determine an initial direction. This involves defining what problems to approach, what the fundamental gameplay mechanics needed are, and what the desired results are. Once possible games have been prototyped, player feedback can begin to be incorporated. Early playtesting helps to uncover what elements of the problem are fun and which can be most confusing and difficult to understand. This can help to both focus the gameplay and narrow the scope of the game to where players will most likely be able to contribute.

After making the game available to the public, a large amount of data and feedback can become available to help improve the game. As in a traditional game, data on gameplay can be gathered from players for an objective analysis of what players are doing, and feedback from the player community is extremely useful in determining new features. However, in a scientific discovery game, as scientists post puzzles and player solutions are analyzed, this analysis must then be incorporated in the design of the game, progressing towards ever better results.

Following this pattern, Foldit has evolved significantly since its initial release. A timeline of significant events in the evolution of the game are given in Figure 3.4.

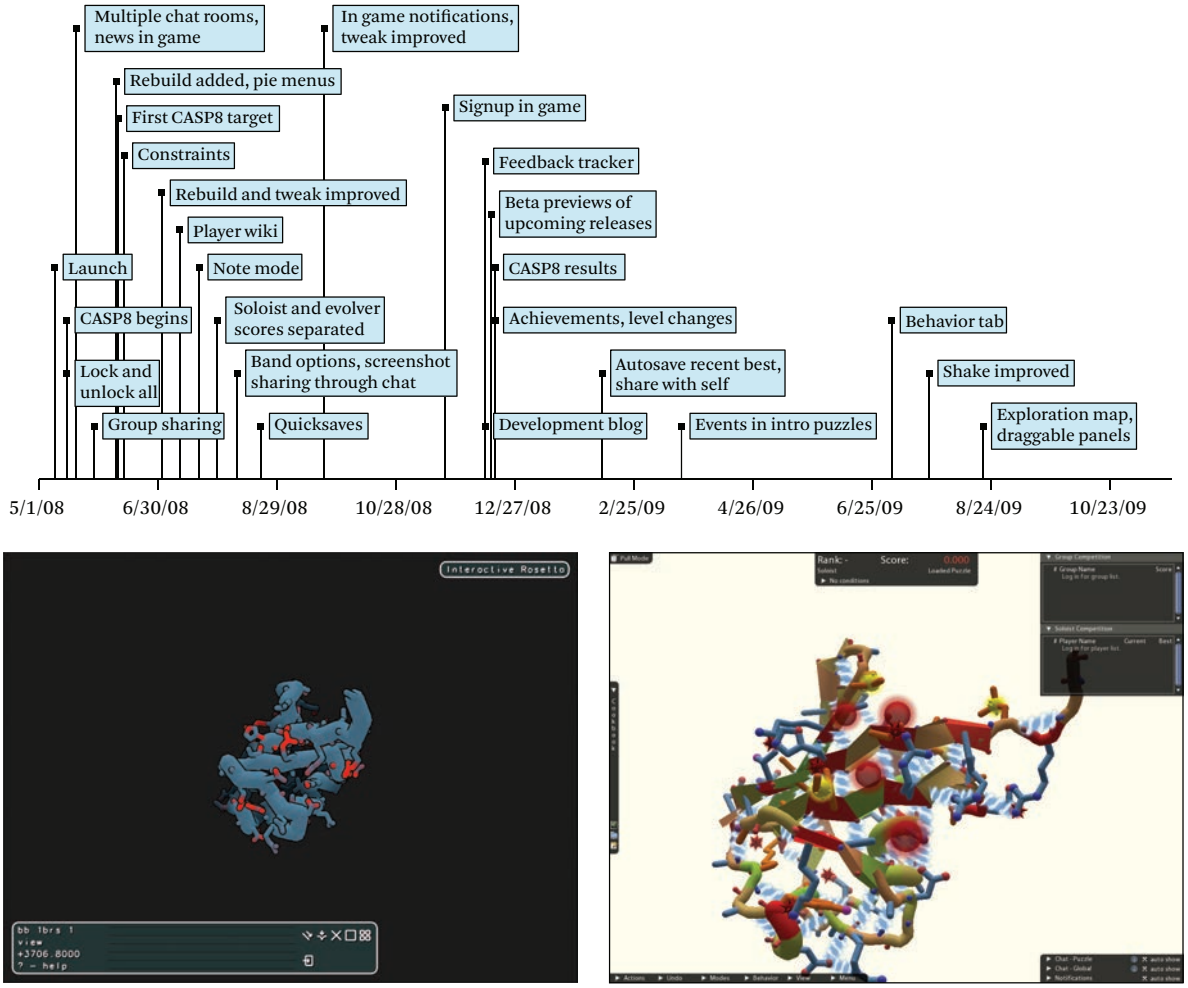

Figure 3.4 **Selected events from the game's evolution over time.** The timeline is shown on the top. Screenshots are included from before release (bottom left) and the current version (bottom right). (Figure from Cooper et al. [2010b])

3.3.3 Categorization as a Game

Although it relies heavily on simulation and visualization, Foldit can be classified as a game, as it possesses the qualities of a game set forth by Schell [Morgan Kaufmann]. Here we list the qualities and how Foldit embodies each.

1. Games are entered willfully: We do not require players to play Foldit.
2. Games have goals: Foldit's goal is to find the best scoring structure.