The Heart of Puzzling: Mathematics and Computer Games

Michael R. Fellows
Department of Computer Science
University of Victoria
Victoria B.C. Canada V8W 3P6
mfellows@csr.uvic.ca

April 21, 1997

Abstract

Mathematics is much more than arithmetic. Modern mathematics is best characterized as the study of form, structure, pattern, process, information and communication. It has ever-widening, astonishing and colorful frontiers and applications. The talk describes how modern mathematical ideas can be tapped to provide not only new kinds of educational games, but novel puzzles and interaction structures for computer games in general. The case is presented that puzzle content and interaction structure are central to all game design, that these issues are inherently mathematical in nature, and that there are many unexplored opportunities for tapping the power of mathematics in computer gaming.

1 Introduction

At the heart of every good game is a puzzle with an "addictive" power of fascination. When this power is present, graphics and story are clearly revealed as important but secondary issues. In action games, aspects of *thinking puzzles* are combined with *dynamic algorithmic puzzles.* Educational games are concerned with presenting puzzles that are sanctioned by their supposed relevance to official school curricula. All three kinds of games are discussed in this paper from the point of view of an emerging *mathematical theory of game design*, and from a point of view that appreciates the affinities between mathematical problem-solving and computer gameplay.

The main points are summarized as follows.

• Puzzle and action structure are the central issue in game design.

• Puzzle and action structure are inherently mathematical issues, and there are many unexplored opportunities for systematically tapping powerful kinds of mathematics for innovative game structures and puzzles.

• Appreciating the mathematical nature of computer gaming is likely to empower new opportunities in education.

This is a somewhat abbreviated version of the paper, due to space limitations for the conference proceedings. The full version (with a number of figures) is available from the web site `http://www.cs.uidaho.edu/~ casey931/mega-`

2 Some Background Information

In this section I will sketch how the point of view presented in this paper has developed. This sketch will include a summary of my talk at CGDC in 1994, and some pointers to potentially useful background sources of information. Because of space limitations, I will need to assume that the the reader has a basic understanding of what modern mathematics is about, and its power and diversity. Some useful sources for such an appreciation can be found in the References. In the technical sections (§3 and §4) I will assume the level of sophistication concerning the mathematical foundations of computing that one would normally be exposed to in an undergraduate degree in Computer Science, and sometimes more.

MEGA-Mathematics and Computer Science Without Computers

I first became involved in bringing mathematics to children by visiting my children's elementary school classrooms. In the beginning (about 8 years ago) the classroom was basically a hippy freeschool in Moscow, Idaho, where there was (delightfully!) no confusion of mathematics and shop-keeper arithmetic. My task was to share some *mathematics*, in the same way that an astronomer might share something of astronomy. The goal of this sort of classroom experience is perhaps best described as *science popularization*.

One of my favorite quotations on this subject is from Kurt Vonnegut, who said:

> If you're going to teach, then you should either teach graduate school or fourth grade.
>
> ...
>
> And if you can't explain it to fourth-graders, you probably don't know what you're talking about.

Typically, I would go into the classroom, and do whatever it was that I was just doing at the university, in the form of a concrete puzzle or activity. [1] Most of the children in our freeschool were about 7 years old then, and we had great fun with topics such as: *The Minimum Weight Spanning Tree Problem*, the computational complexity and combinatorics of *Map and Graph Coloring*, the combinatorics of strings of symbols (nowadays relevant to DNA analysis), *Knot Theory*, *Orientable and Non-Orientable Manifolds*, *Error-Correcting Codes*, and *Cryptography* (including zero-knowledge protocols and combinatorial public key schemes). Many of the mathematical topics native to Computer Science are marvelously concrete and active. An algorithm, after all, is an activity. Keep this in mind, because it means that *a computational problem is a puzzle*, rightly viewed.

An account of these early experiences can be found in [Fe93]. FFrom these beginnings developed a number of current projects:

• The C-3 Group at Los Alamos became interested, and through the inspired leadership of Vance Faber and Bonnie Yantis began the the MEGA-Math program to develop materials for teachers, and to make these materials freely available on the World-Wide Web.

• Neal Koblitz of the Mathematics Department at the University of Washington became interested and we have had wonderful experiences doing computer science without computers (and sometimes without even electricity) in Peru, Vietnam and rural El Salvador in conjunction with the Kovalevskaia Fund. [2] See [?] and [Ko96] for some discussion of the important issue of the *Computer Cargo Cult* in education.

[1] Lately, this is computational biology. Some elementary classroom materials about this are available from the author [DFW96].

[2] The Kovalevskaia Fund has been involved for more than ten years in encouraging and supporting women in science in Third World countries, including Vietnam, Nicaragua, El Salvador and Peru. For information, contact Ann Koblitz `koblitza@hartwick.edu`.

• In British Columbia, the classroom visits have blossomed into the larger activities of the Mathmania Society, supported by a B.C. provincial grant, and working with other organizations such as Science World (the children's science museum in Vancouver), Family Math (of the Lawrence Hall of Science in Berkeley), and the upcoming CRYPTO '96 meeting [3]. A *Family Mathmania* that was held recently (on a Friday night, even) at a Victoria elementary school, 6:30 – 8:00 pm in the gymnasium, attracted a crowd of over 200 parents and children (the local organizer expected less than 50). We Mathmaniacs now have a routine where we switch back-and-forth between puzzles on paper, and huge tarps on which sorting networks, finite-state machines, etc., are done in colored hockey tape, providing a kind of theatre space for big physical mathematical activities and games.

• On several visits to elementary schools I mentioned open (i.e., unsolved) mathematical problems that I thought were "kid-sized" enough to offer a prize for a solution. Some of these problems are described in Chapter 4 of [CaFe93]. The prize was a lunch-date and a Mathmania T-shirt. It was *amazing* (and touching) how kids threw themselves into working on these problems! This has led to a joint project of Mathmania and the Canadian Mathematics Society to create a web site of kid-sized open mathematical problems with prizes (with funding by Industry Canada and the Province of British Columbia). The web address is: `http://www.csc.uvic.ca/~ mmania`. The project is called *Erdös for Kids*.

An Educational Revolution Via Computer Games?

¿From the beginning, those of us in the community of researchers in theoretical computer science who have perceived the marvelous opportunities for math education in the new content material of Computer Science have been interested in the idea of using computer games as a means to deliver this good stuff.

My talk at CGDC in 1994 is summarized as follows:

(1) By means of many examples, and some discussion of the mathematical ideas [CaFe96] that can be engaged in open-ended problem-solving based on material such as *Coloring*, *Knot Theory*, *Finite-State Machines*, *Sorting Networks* and *Cryptography*, I tried to make the point that there is a wealth of engaging mathematics that *could* be presented to kids in games, much of it native to Computer Science.

(2) I complained about the dismal state of affairs in computer games for mathematics education, much of which is little more than arithmetic drill decorated with "cute" animation. Boring! And what an ironic situation, given the wealth of mathematical topics associated with computing and nowhere to be seen in these products *based* on computing.

(3) I tried to point out that the current semi-revolution in mathematics curriculum that centers on the NCTM Standards [NCTM89] would seem to open the door wide to educational games based on *real* mathematics (not just shop-keeper arithmetic), and that the computer games industry could play an exciting role in education reform by seizing this opportunity to innovate.

Looking back over the past two years, one can see that this simply hasn't happened. In the realm of education, the computer games industry has proved to be remarkably timid and conservative.

And Now: Breaking Out of the Educational Ghetto?

I make the following predictions about what's going to happen next.

Prediction 1: The spirit of conscious and systematic attention to Mathematics and the possibilities for mathematical play is going to escape from the ghetto of educational games and infuse *all* of the world of computer game design.

[3] Any interested person living in the vicinity of Santa Barbara please contact me.

Prediction 2: After this happens, we will finally begin to see computer games live up to their liberating potential in education.

Prediction 1 is a very safe bet. Almost any prediction of the form "X is going to become more mathematical" has been right on, all the way back to Pythagoras. Games and gameplay have remarkable natural affinities with mathematics that are worth noting:

• Games are *syntactically rigid.* Once you figure out that a drop-kick, jump-twist-jab kills the Megamonster, all Megamonsters have suddenly become trivial. You have essentially discovered a formula that is completely dependable and syntactically reliable — it is in the nature of the computer code for the game that this is so. This is different from monsters in "real life" but it is *a lot* like Mathematics. If beating the game is the theorem, this is a lemma along the way. Experienced players of games learn to make reasonable conjectures, and in general behave like research mathematicians (albeit, unconsciously).

• If you haven't got a puzzle, basically, you haven't got a game. The addictive fascination of computer game-playing has a fundamental and remarkable similarity to the subjective mental experiences of mathematical research. The rest of this paper presents the argument that systematic connections can be made between these two realms.

3 Ten Thousand Puzzles: A Method

One of the most important working reference books in theoretical computer science is the book on *NP*-completeness by Garey and Johnson [GJ79]. Those of us working in the areas of algorithm design and computational complexity routinely refer to this book in its role as a compendium of computational problems, and what is known of their complexity. Much of the significance and intellectual drama of theoretical (i.e. mathematical) computer science is represented in the elegant and fundamental concept of the complexity class *NP*, and in the thousands of important workaday problems that are known to belong to this class.

The reader unfamiliar with this body of knowledge should at least be made aware that "all" of the computational dramas of all kinds of situations in the world are cataloged here, stripped down to their mathematical essentials, but often retaining in their names some hint of their varied and colorful practical origins: THE TRAVELING SALEMAN PROBLEM, THE CHINESE POSTMAN PROBLEM, THE MAP COLORING PROBLEM, BIN PACKING, STOPLIGHT SCHEDULING, THE PIANO MOVERS PROBLEM, etc. Below we will describe some simple and representative problems from the class *NP*.

The main point of this section is to describe and give examples of how to use this compendium to systematically generate "endless" numbers of amusing and challenging puzzles. Basically, there are two steps in this process.

Step 1: Identify an information-action *handle*.

Step 2: Attach this handle to the *NP* Catalog and turn the crank. (Repeat as many times as desired.)

We will illustrate by using a simplified *NP* Compendium that consists of the following two problems.

MINIMUM DOMINATING SET
Instance: A graph $G = (V, E)$ and a positive integer k.
Question: Is there a set of k vertices $V' \subseteq V$, such that $\forall v \in V \exists u \in V'$ with $u = v$ or $uv \in E$?

FEEDBACK VERTEX SET

Instance: A graph $G = (V, E)$ and a positive integer k.
Question: Is there a set of k vertices $V' \subseteq V$, such that every cycle in G contains at least one vertex of V'?

The Discreet Repairs Handle

This handle is modeled to some extent on the information drama of the game Minesweeper. The essence of the problem is that there is a hidden flawed solution that needs to be repaired, but without exposing the flaws. This is a sufficient general puzzle schema that it can be applied to *any* problem in *NP*.

Discreet Repairs for DOMINATING SET

Have a partner draw a graph, and mark on on it a "flawed" dominating set. By this we mean a dominating set that dominates all but one vertex. Have your partner cover each vertex with a penny, and begin the puzzle. At the beginning of the play, you are awarded a single token. On each play, you can either: (1) uncover a vertex byremoving a penny, or (2) play your one token (which effectively selects one additional vertex for the dominating set). If at any time you have uncovered all of the neighbors of an uncovered vertex v and none of these vertices is in the hidden V', then a fatal flaw has been revealed and *you lose!*

Discreet Repairs for FEEDBACK VERTEX SET

This is much the same. Have your partner mark a feedback vertex set that fails on exactly one cycle. As in the last puzzle, you have one token with which to make a repair. If at any point a cycle has been revealed, you lose.

The On-Line Problem Handle

Deep in the heart of Tetris, from a mathematical point of view, is an *on-line two-dimensional packing problem*. The subject of algorithm design for on-line computational problems — where the solution must be constructed as the instance is revealed — has become one of the hot areas of research in theoretical computer science in the last six years. There are frequently entire sessions at major conferences devoted to on-line algorithmics and complexity theory. With this perspective one can create a "Tetris of X" where X is any problem in *NP*.

For a simple example, Tetris of Dominating Sets has a graph scrolling downward and the player has some (limited but replenished) supply of tokens that can be played to indicate a dominating set. If an un-dominated vertex hits the bottom of the screen, *you lose!*

Other Handles

There are many other handles that can be attached to *NP* to systematically generate puzzles. Cryptography seems to be a particularly rich source. One would expect at least several dozen distinct handles that are "universally applicable" to problems in *NP* to be identifiable in any systematic inventory. The main point of this section is simply that concrete complexity theory is a goldmine of puzzle material that can be methodically tapped in various ways.

What About the Storyline?

Computer games are not just puzzles, they are also stories. If we have an interesting abstract puzzle, how do we attach a story to it? The important point is that *all* of the problems in the *NP* compendium of [GJ79] are there because they arise in at least one, and perhaps many, natural and important ways in the world. Recall what these are, and you have obvious leads on storylines.

4 Towards New Game Engines and Gameplay Structures

Mathematical structure is relevant to game design in other and deeper ways than the simple puzzles considered in the last section.

Fractals and Graphics

The use of fractals to generate images is now widespread. Many of the graphic images of the game Myst were generated in part by fractal graphics techniques. So here is one possibility: fractal methods are simply used to *generate* some small number of fixed images for the game. One might similarly employ the related mathematical methods of L-systems and stochastic grammars.

A more exciting possible use of these techniques is to incorporate into the game efficient algorithms for *generating* the images during the gameplay, thus providing an effectively infinite variety of images. [4]

L Systems and Sound Scores

One of the annoying characteristics of most computer games is that the musical scores are repetitive and simplistic. Essentially, this comes down to the problem that the score consists of a fixed number of individually crafted soundtracks. Using L-systems, one can generate in nearly real-time an "infinite" variety of interesting scores [Vic96].

Genetic Algorithms and Enemies

In a game such as Space Invaders the player is confronted with a small fixed number of kinds of "enemies", each of which is handcrafted and therefore has a fixed behaviour. Drawing on his research expertise in genetic algorithms, Ian Parberry of the University of North Texas has designed a game where the enemies "evolve" during the gameplay — so that set of kinds of enemies and enemy behaviours is potentially infinite, even in a simple game that is relatively easy to code [Par96].

Finite and Infinite Game Spaces

A frequently noted annoyance in gameplay is the feeling of claustrophobia that comes when you realize that there are just 100 places to be in the game (especially when you are forced to trek repeatedly back and forth through these same few locations searching for the hidden door). In its mathematical essentials, you are basically stuck in a deck of hypercards. Each panel has a small number of action *buttons*. Your game quest consists of choosing an action (a button), which takes you to a new card. Many games are prototyped in hypercard, and all such games are basically a finite-state maze. [5]

What are the possibilities for games that involve a potentially infinite number of situations, or that have a more complex underlying structure than a finite-state maze? There are several possibilities that appear to be completely unexplored presently.

Finite-state machines are used to model an amazing variety of situations and issues in computing, but there are some areas where they are just too simple, and more complex mathematical tools have emerged. One of these areas is that of concurrent communicating processes (as in operating systems) and distributed computation (as in computer networks). One of the standard mathematical tools used in these areas is the notion of a Petri net. Petri nets have been used to model an amazing variety of concurrent computational issues, including cognitive dynamics in mathematical problem-solving [DeB96]. Petri nets are an obvious can-

[4] There is an article about a program that accomplishes this in the January, 1996, issue of *Scientific American*.

[5] Indeed, if you wanted to explain to a non-mathematician what a *finite-state machine* is, you could begin by explaining that it is essentially like a typical computer game — an observation due to Frank Ruskey.

didate for more complex game structures and more powerful prototyping and design tools (beyond hypercard systems).

The issue of finite versus infinite (but still meaningfully structured) sets is a deep theme in abstract algebra. Correspondingly, we can find in this branch of mathematics some candidate tools for constructing infinite games. One of these is to use an infinite confluent rewrite system [?] to create a kind of virtual infinite deck of hypercards. The rewrite system would manage an abstract index to describes each card in the deck and its buttons. (Confluent rewrite systems are closely related to presentations of abstract groups.)

Necessary for both the Petri net and confluent rewrite system approaches to this problem are efficient algorithms for producing the visual panel from information that points into the game space (i.e., from an abstract representation or index). Note that this issue arises in all of the above discussions. One would expect efficiency issues to be crucial in harnessing these mathematical techniques for games. There might be certain tradeoffs. In a quest game, the story is highly crafted and deliberate, but the game space finite. In an infinite game space, you might have a storyline that is somewhat out-of-control. Maybe Alice would know what to do. [6]

Chasing Through Manifolds: Games and Topology

In a chase-and-shoot spaceship game, one could easily modify the code so that the spaceship is flying around in, say, a non-orientable 5-manifold, with 2 dimensions tightly coiled, and where you are considered cloaked by mirror-images. Or perhaps the normal 4-manifold S^4 with the homotopy class of the Hopf map playing a role in the operation of the warp drive. Putting some algebraic topology into spaceship shoot-and-chase does not pose *any* significant programming challenges whatsoever. The fact that all these games are currently being playing in "boring old Euclidean space" simply reflects a lack of imagination among designers. Manifolds are *not* a difficult mathematical idea. The representation on a computer screen of location information in higher dimensional manifolds is easy and natural. A gentle and visual introduction to manifold theory can be found in [We85]. *Kids will dig non-Euclidean topological spaces and game dynamics for the same reasons that topologists find them fascinating.*

5 Games and Education: The Future?

I mentioned in §2 that at the recent Family Mathmania we had a wonderful time with the elementary school children puzzling about finite-state machines implemented in colored hockey tape on giant tarps. The Mathmania student volunteers (from the university) who were helping came from two different groups:
(1) Undergraduates who were in my Mathematical Foundations of Computer Science course at the University of Victoria last semester. The theory of finite automata is a major part of this standard ACM curriculum course.
(2) Graduate students. I currently supervise three Ph.D. students working in the exciting area of computational biology. It turns out that finite automata are extremely important to our research.

This confluence of experiences and interests has led to the formulation of the following two "scenarios of the possible".

The Movie

We have developed a plot premise for a movie that is *sort of* a sequel to Jurassic Park, and simultaneously *sort of* a sequel to The Net.

[6] Or the Canadian poet Jim Andrews, who has written of Alice's adventures in an infinite mathematical space.

Fact: If one could find a small collection of finite automata (perhaps multiheaded or having bounded depth stacks) that by their aggregate behaviour on a DNA sequence signalled its functional significance, the computer disk on which you record this collection of automata ("infosomes" — a kind of virtual ribosome) would be worth hundreds of millions of dollars to the drug companies bent on harvesting commercial gain from the Human Genome Project.

Plot Premise: The hungry and desperate graduate students, led by their morally ambiguous professor with gambling debts, realize that "culturing" such a collection of automata requires a lot of ad hoc problem-solving ingenuity. They create a computer game that harnesses the cleverness of 12 year-old children to this labor. The game has a fascinating addictive power (partly because it is based on puzzles that embody this research frontier) and thousands of children who play the game put in hundreds of thousands of hours of child-mind-fresh effort. The game is "new generation" involving hot-links to the Web. (This is slightly in the future, where net connections are routine.) The game incorporates a secret watchdog routine that reports (to the creators of the game) any progress that has been made in finding effective infosomes.

OK, maybe it'll never fly with Hollywood. But unless you right on the cutting edge in computational biology, you have no idea how plausible this is!

The Textbook

At the end of the Family Mathmanias we have often wished that we could offer the children a computer game disk to take home, so that they could continue to puzzle and play with finite automata or sorting networks or the complexity of coloring. David Vogt, the Director of Science at Science World in Vancouver, has expressed similar wishes, as Science World is attempting to become in part a Virtual Science Museum in order to better serve the children of this geographically far-flung Province. Such games could be "connected" to the Erdös-for-Kids web site, where open mathematical problems accessible to child research are described. Much of our Family Mathmania material is native to the curriculum for the Mathematical Foundations of Computer Science course.

Why not write a "textbook" that serves the entire community that we see at a Family Mathmania? The textbook would consist of a series of 8-10 games (e.g., the "NFA-to-DFA" game) where: difficulty levels 1-4 are fun for the elementary school age group: it's impossible to beat level 8 without understanding the Myhill-Nerode theorem (i.e. you have earned at least a B in the university course), and level 10 — well, that's where the watchdog routine kicks in!

References

[BWF96] T. Bell, I. Witten and M. Fellows. *Computer Science Unplugged ... offline activities and games for all ages.* Manuscript (searching for a publisher), 231 pp., 1995.

[CaFe93] N. Casey and M. Fellows. *This Is MEGA-Mathematics!* Los Alamos National Laboratories, 134 pp., 1993. Freely available from `http://www.c3.lanl.gov/~ captors/mega-math`.

[CaFe96] N. Casey and M. Fellows. "Let's Focus on the First Four." To appear in: *Discrete Mathematics in the Schools: How Can We Have an Impact?* (D. Franzblau, F. Roberts and J. Rosenstein, eds.) DIMACS and the American Mathematical Society, 1996.

[Ci93] B. Cipra. *What's Happening in the Mathematical Sciences, Vol.1, 1993.* American Mathematical Society (800-321-4267), Boston, 1994, (ISBN 0-8218-8999-0).

[Ci94] B. Cipra. *What's Happening in the Mathematical Sciences, Vol.2, 1994.* American Mathematical Society, Boston, 1995, (ISBN 0-8218-8998-2).

[Ci95] B. Cipra. *What's Happening in the Mathematical Sciences, Vol.3, 1995.* American Mathematical Society, Boston, 1996, (ISBN 0-8218-0355-7).

[DeB96] V. DeBellis. *Local Emotional Dynamics in Mathematical Problem-Solving*, Ph.D. dissertation, Department of Education, Rutgers University, 1996.

[DFW96] V. DeBellis, M. Fellows and K. Wood. ???

[Fe93] M. Fellows. "Computer SCIENCE and Mathematics in the Elementary Schools," In: *Mathematicians and Education Reform 1990–1991.* Conference Board of the Mathematical Sciences, *Issues in Mathematics Education* vol. 3 (1993), 143–163. (Available from the MEGA-Math `ftp` site.)

[Ga82] M. Gardner. *Aha! Gotcha.* W. H. Freeman and Co., New York, 1982.

[GJ79] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness.* W.H. Freeman, San Francisco, 1979.

[KKF95] A. Koblitz, N. Koblitz and M. Fellows. *Notices of the American Mathematical Society*

[Ko96] N. Koblitz. *The Mathematical Intelligencer*

[NCTM89] *Curriculum and Evaluation Standards for School Mathematics*, National Council of Teachers of Mathematics, Reston, Virginia, 1989.

[Pa91] J. A. Paulos. *Beyond Numeracy.* Alfred A. Knopf, New York, 1991.

[Par96] I. Parberry, private communication, January, 1996.

[Pe88] I. Peterson. *The Mathematical Tourist: Snapshots of Modern Mathematics.* W. H. Freeman and Co., New York, 1988.

[Vic96] The L-Systems Music Project, University of Victoria, 1995.

[We85] J. R. Weeks. *The Shape of Space.* Marcel Dekker, Inc., New York, 1985.