



On the Complexity of Katamari Damacy

by [Gregory M. Zaverucha](#)

Introduction

The complexity of algorithm-based games has been of interest to computer scientists for a long time. An early text focusing on mathematical games by Berlekamp, Conway, and Guy [2], *Winning Ways for your Mathematical Plays*, has since grown into a four volume collection (by the same title). Much research has gone into classic puzzle games, for both one and two players. A survey of results of classic games is Demaine [5], who calls it *algorithmic combinatorial game theory*; do not mistake this with *economic game theory*, which deals with things like auctions. Examples of classic games that have been studied include chess, checkers, and go. Some examples of computer games with known complexity include Minesweeper [8], Tetris [6], and Lemmings [6]. The types of games that typically receive attention are single player puzzle games, two player board games, and mathematical games.

The study of a game such as Katamari Damacy is somewhat novel because games of this genre have received little attention. Just exactly what category of game Katamari Damacy belongs to is unclear. It does share some of the properties of addictive puzzle games. It is simple to understand and start playing, but it takes work to excel at. However, the game-play and style are quite different from typical puzzle games. Wikipedia refers to it as a "puzzle-action" game [11], and IGN calls it a "third-person action" game [7]. Whatever its genre, Katamari Damacy is a break from the conventional games typically studied by computer scientists.

Many games that have stood the test of time are very difficult to play algorithmically. A common thread of these games is that they are instances of NP-hard (nondeterministic polynomial-time hard) or NP-complete problems. These games are challenging,

although tractable, for humans. Sudoku is a good example of this because the generalized problem (with arbitrary dimensions) is NP-complete [1]. In this respect, Katamari Damacy is not so different.

Commercial Details

We reproduce a few details from [11]. Katamari Damacy was designed by Keita Takahashi and then developed and published by Namco. The game was first released for the PlayStation 2 console in Japan on March 18, 2004 and in North America in September of the same year. It has both one and two player modes, although this article focuses solely on the single player game. Both the Computer Entertainment Rating Organization (the Japanese video game rating association) and the Entertainment Software Rating Board (the American video game rating association) gave the game a rating meaning that the game is suitable for players of all ages. The game received multiple awards for both its innovative design and its soundtrack.

Game Storyline and Description

The game's story is summarized in the game manual [9] as follows:

In a freak accident, the King of All Cosmos inadvertently annihilated all the stars. The task of fixing the King's mistake has fallen upon his son, the Prince. In order to restore the glorious starry sky, the Prince must roll around a heap of objects on Earth, gathering more and more things from the item-rich planet and sending them off into the night sky. Does the Prince have what it takes to succeed?

The heap of objects is named the *katamari*, which translates roughly to "clump" from Japanese. *Damacy* translates to "spirit" [11]. When the Prince begins a level, the katamari is small. The player controls the Prince, who pushes the katamari to make it roll around. Small objects will cling to it, and it grows in size. As the katamari grows, larger and larger objects will cling to it as well. Eventually people, elephants, buildings, and trees can be collected. Although there are other types of levels, this article focuses on the type known as a "Make a Star" level. Each "Make a Star" level has two goals. First, the katamari must reach a diameter specified by the King before the timer expires. If there is time to spare after reaching this size, the second goal is then to make the katamari as large as possible. When the timer expires, the King brings the Prince and the katamari back to space with the Royal Rainbow. If the katamari is large

enough, the King creates a star from it; and if the first goal was reached quickly, a comet appears in the sky as well.

This begs a question: for the Prince to build a star of size n , how long should the King of All Cosmos expect to wait?

Intended Audience

Hopefully, this article will be enjoyed by as many fans of the game as possible. To this end, the presentation will be kept simple, and judicious use of informal arguments has been made. Ideally, this article will be accessible to undergraduate readers who have had some exposure to complexity theory. The main section of the article will rely on results from the common undergraduate textbook of Cormen, Leiserson, Rivest, and Stein [3]. Therefore, instructors using this text may find that this article makes an interesting example to supplement those given in chapters 34 and 35.

Formal Description

In this section, the terms "player" and "algorithm" will be used synonymously because using an algorithm to play the game can be viewed as simulating an automated player ("computer player," in video game terminology).

In order to work with the game, some simplifying assumptions must be made. Each of the assumptions, however, will only make the game easier for players. It will then be shown that playing the simpler version of Katamari Damacy is NP-hard, and then it will be demonstrated that the actual (more difficult) game is hard as well.

The first simplifying assumptions are that the landscape is a 2-dimensional plane and the player's motion within the plane is unrestricted. The latter assumption will be termed the *unrestricted motion assumption*. All distances in the plane are the standard Euclidean distance. An instance of the game, G , will be defined as a collection of "subgames." Each of the subgames correspond to the katamari's current size, which is represented as an integer in the range $1, \dots, s$. The katamari starts small, so the initial size is 1. If the player collects all of the objects in the game, the katamari would have size s . So an instance, G , is divided quite naturally into the series of subgames, G_1, \dots, G_s .

Furthermore, it is assumed that at subgame G_i , the only objects in the environment

are those that the player can collect (i.e., objects of size i or less). Let T_i be this set of objects, and represent each object by a point in the plane. Finally, let t_i be the cardinality of T_i .

A further assumption on the game can now be made, which will be called the *perfect information assumption*. With perfect information, for G_i , the player is assumed to have complete knowledge of the set T_i . The player completes G_i by collecting all of the objects in T_i . But this should be simple! Given a set of points in a plane, the player must plan a path that includes every point. Indeed, if the game were this simple, it would certainly not have enjoyed the success it has.

The player does not have to collect *all* of the objects at a certain size, i . Instead, once a sufficient number of items have been collected, the player can move on to size $i+1$. To deal with this, it might be formally required that the player need only collect k_i out of t_i objects. Which k_i are collected is left up to the player.

The game has two goals, or criteria that are used to evaluate the player's performance. The names of the goals are inspired by the game.

1. **The comet goal:** Reach size x in minimal time. A game is optimally played with respect to the comet goal if the time required to reach size x is the least time possible, T_{min} .
2. **The star goal:** Grow the katamari as large as possible within the time limit. With respect to the star goal, play is optimal if the katamari is the maximum possible size, S_{max} , when the timer expires. If time allows the player to pick up all of the objects in G and if all objects are collected in the least amount of time, then G is played optimally.

Complexity

In this section, it will be shown that playing Katamari Damacy optimally with respect to either of the goals in the previous section is NP-hard. The next lemma will form the crux of this proof (Theorem 3.7).

Lemma 3.1 Let Q be a player who plays G optimally with respect to either the star goal or the comet goal. Q is able to find a path through k_i of t_i objects from T_i of minimal distance (for all G_i).

Proof. Recall that it is assumed that the katamari rolls at constant speed. Suppose that for some G_i , the path followed to collect the required k_i objects, call it P , is longer than the minimal path, P_{min} . Since the speed is constant, longer paths require more time to follow. Let Q' be identical to Q except Q' uses P_{min} in G . In the case of the comet goal, Q' finishes faster than Q . For the star goal, there are two possibilities. If everything in the world is picked up, then Q' requires less time than Q . If not all objects are picked up, Q' will be larger than Q because it has time to pick up more objects. In any case, Q' outperforms Q , which is a contradiction because Q is optimal. Therefore, an optimal player must use shortest paths. \square

At this point, most computer scientists should have recognized that optimally playing a subgame, G_i , is closely related to the *traveling salesman problem* (TSP). We start with the definition of the TSP.

Definition 3.2 *The traveling salesman problem (TSP)*

INPUT: A graph, $G = (V, E)$, with n vertices and an $n \times n$ matrix, D , where $D_{i,j}$ gives the distance (or cost) between vertex i and j .

OUTPUT: A Hamiltonian cycle through G of minimal length, i.e., a path through G that includes all n vertices and is the shortest possible such path.

The TSP is a well known NP-complete problem. It will also be important to know it can be approximated, which is established with the following two theorems. Recall that an *r-approximation algorithm* produces a solution that is never more than r times worse than the optimal solution. The parameter, r , is called the *approximation ratio*. Approximation algorithms are the subject of [3, Chapter 35]. The *triangle inequality* (in a geometric setting) essentially says that the shortest path between two points is always a straight line.

Theorem 3.3 There exists a polynomial time 2-approximation algorithm for the traveling salesman problem with the triangle inequality.

Proof. See [3, Algorithm "Approx-TSP-Tour"]. \square

For more precise results about how close an approximate solution can be to optimal (for instances with the triangle inequality), see the paper of Papadimitriou and Vempala [10]. Without the triangle inequality, there is no polynomial time

approximation algorithm.

Theorem 3.4 ([3, Theorem 35.3]) If $P \neq NP$, then for any constant, $r \geq 1$, there is no polynomial time approximation algorithm with approximation ratio, r , for the general traveling salesman problem. \square

Earlier, it was pointed out that the player does not necessarily have to collect *all* of the objects in a particular subgame. The player can choose to roll up k_i of t_i objects in T_i for $k_i \leq t_i$ (k_i must be greater than some minimum number of objects required to make the katamari large enough to proceed to the next subgame). Once the player has decided which k_i objects to collect (using an oracle or otherwise), he or she is faced with the following problem:

Definition 3.5 *The k of n traveling salesman problem, $((k,n)$ -TSP)*

INPUT: A graph, $G = (V,E)$, with n vertices, an integer, $k \leq n$, and an $n \times n$ matrix, D , where $D_{i,j}$ gives the distance (or cost) between vertex i and j .

OUTPUT: A path through k vertices of G of minimal length.

The (k,n) -TSP generalizes the problem in Definition 3.2, which is the special case, (n,n) -TSP. It is now shown that this variant of the traveling salesman problem is no easier than the original.

Theorem 3.6 The (k,n) -TSP is NP-hard.

Proof. Suppose the (k,n) -TSP can be efficiently solved when k is some fixed fraction of n , e.g., $k = \text{floor}(n/d)$. Given an instance of TSP with n vertices, it can be solved in the following way. Let m be the maximum distance between any two of the original n vertices. Let $n' = dn$, and add $n' - n$ vertices, such that the distance between any new vertex and any of the old vertices is larger than m . Furthermore, the distance between any two of the new vertices should also exceed m . Thinking in the 2-D case, the vertices in the TSP instance to be solved form a "clump," and the vertices added are "out at the horizon in different directions." There is an n of n' instance that can be solved to get the optimal solution for the TSP instance. Therefore, solving the (k,n) -TSP is at least as difficult as the TSP. \square

Suppose the unconstrained motion assumption is removed and that the environment is no longer modeled by a 2-D plane. As a result, the player's movements must follow the

paths allowed by the game controls and Katamari physics. Furthermore, the player's motion is limited both by larger objects and the terrain. Therefore, the triangle inequality no longer holds in this case.

In absence of the triangle inequality, there is no polynomial time approximation algorithm with approximation ratio r for the (k,n) -TSP. By this same reasoning, as in the proof of Theorem 3.6, the existence of such an algorithm would mean that the general TSP could also be approximated in polynomial time with approximation ratio, r , which is not possible by Theorem 3.4. Summing up this section, Theorem 3.7 describes the complexity of Katamari Damacy.

Theorem 3.7 Optimal Katamari Damacy is NP-hard. If $P \neq NP$, there is no polynomial time algorithm that can play Katamari Damacy optimally. Furthermore, for any constant, r , there is no polynomial time approximation algorithm with approximation ratio, r , for optimal Katamari Damacy.

Proof. By Lemma 3.1, an optimal player must solve the (k,n) -TSP, which is NP-hard by Theorem 3.6. Hence, Katamari Damacy is NP-hard, and it cannot be played optimally in polynomial time. An approximation algorithm for optimal Katamari Damacy would also approximate the (k,n) -TSP and TSP problems. Therefore, optimal Katamari Damacy cannot be approximated in polynomial time either. \square

Conclusion

This article has shown that optimal Katamari Damacy is NP-hard by reduction from a variant of the traveling salesman problem. One could also ask whether optimal Katamari Damacy is NP-complete. The difficulty of optimally choosing the value, k , and then choosing a set of k objects to collect would also have to be dealt with. Algorithmically playing the game in two player mode would also make for interesting future work.

Acknowledgments

The author thanks the following people for reviewing an earlier draft of this article: Chris Hamilton, Kevin Henry, Glenn Hickey, Abram Hindle, and Aniket Kate.

References

1

L. Aaronson. Sudoku science. *IEEE Spectrum*, 43(2):16--17, February 2006.

2

E. R. Berlekamp, J. H. Conway, and R. K. Guy. *Winning Ways for your Mathematical Plays*. Academic Press, London, 1982.

3

T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. McGraw-Hill, Boston, 2nd edition, 2001.

4

G. Cormode. The Hardness of the Lemmings Game, or "Oh no, more NP-completeness proofs". *Technical Report 2004-11*, Center For Discrete Mathematics and Computer Science, Rutgers University, May 2004.

5

E. D. Demaine. Playing games with algorithms: Algorithmic combinatorial game theory. In *Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science, pages 18--34. Springer-Verlag, 2001.

6

E. D. Demaine, S. Hohenberger, and D. Liben-Nowell. Tetris is hard, even to approximate. In *Computing and Combinatorics*, Lecture Notes in Computer Science, pages 351--363. Springer-Verlag, 2003.

7

IGN: Katamari Damacy. Accessed April 2007. URL: <http://ps2.ign.com/objects/606/606672.html>.

8

R. Kaye. Minesweeper is NP-complete. *Mathematical Intelligencer*, 22(2), 2000.

9

Namco Bandai Games Inc. *Katamari Damacy instruction booklet*, 2004. Distributed with each copy of the game.

10

C. H. Papadimitriou and S. Vempala. On the approximability of the traveling salesman problem. *Combinatorica*, 26(1):101--120, 2006.

11

Wikipedia. Katamari Damacy. Accessed April 2007. URL: http://en.wikipedia.org/wiki/Katamari_damacy.

Biography

Greg is a PhD candidate at the David R. Cheriton School of Computer Science, University of Waterloo.