# Zen Puzzle Garden is NP-complete

Robin Houston[a], Joseph White[b], Martyn Amos[c]

[a]*mySociety Ltd., London, United Kingdom.*
[b]*Lexaloffle Games, Wellington, New Zealand.*
[c]*School of Computing, Mathematics and Digital Technology, Manchester Metropolitan University, United Kingdom.*

*Keywords:* Computational complexity; NP-completeness; Puzzle.

## 1. Introduction

Zen Puzzle Garden (ZPG) is a one-player puzzle game that takes place on a two-dimensional grid of squares, called a *garden* [1]. Squares may be either *sand*, *rock*, or *walkable*. The objective of the game is to move a *monk* character around the garden, causing him to cover *all* sand squares. The following rules apply:

- The monk may move freely on walkable squares.
- The monk may only move within the von Neumann neighbourhood (i.e., no diagonal movements are allowed).
- Moving onto and then vacating an un-raked sand square causes that square to be permanently raked. The monk may not move from a walkable square onto an un-raked sand square and then immediately back onto the *same* walkable square (that is, sand squares act as one-way gates).
- Once moving on sand, the monk continues to move in a straight line until he encounters either a walkable square (in which case he moves onto it) or a a raked square (in which case he stops moving). The monk may not turn corners during a single move while moving on sand. These two rules are central to the challenge of the game – if the monk could be moved over the sand on a square-by-square basis then most boards would be trivial to solve.

An example garden is depicted in Figure 1, along with a sample solution. Although the problem has been studied experimentally [2], until now no formal proof of its complexity has existed. We now demonstrate that deciding the solvability of ZPG is NP-complete.

**Theorem 1:** Deciding the solvability of a Zen Puzzle Garden instance is NP-complete.

Following [3], we construct a reduction from the following NP-complete problem [4]; given a cubic planar graph, does it contain a Hamiltonian circuit? That is, we construct ZPG gardens which correspond to arbitrary cubic planar graphs, and any garden we construct will have a solution iff the corresponding graph has a Hamiltonian circuit.

## 2. Reduction

To build a ZPG garden that corresponds to a cubic planar graph, $G$, we first draw the graph on a grid. This may always be done in such a way that the area of the grid is quadratic in the size of the graph [5]. We then convert the grid into a ZPG garden as follows. Each square of the grid is converted into a $7\times 7$ tile, made up of either rock, sand or walkable squares. Each tile is either a straight edge, a corner, or a node of the graph with three incident edges. The first two tiles are depicted in Figure 2, and these may be rotated as required.

The node tile is constructed in such a way that, whichever sand square is covered first, it is always possible to cover the remaining sand squares and come out either of the other sides (Figure 3).

In Figure 4 we show a cubic planar graph, along with its ZPG representation. The start node is coloured in red, and the Hamiltonian cycle is shown in bold.

In the ZPG representation, the start node is encoded as a two-tile complex (located at top-center). The top tile (marked with a red dot) is made entirely of blue walkable squares, and is where the monk is initially placed. The lower tile is a single "gateway" tile, depicted in Figure 4 (bottom). This tile is a modified node tile with two of the upper-most rocks converted to sand. The first sand square covered *must* be one of these two, and the last sand square visited must be the other. This facilitates exit from and return to the start node. Any solution must pass through all other nodes only once, so any solution to the garden gives a Hamiltonian circuit though the graph.

We know from [5] that a planar graph on $n$ nodes can be realised using a grid of area $O(n^2)$, and that this grid realisation can be computed in time $O(n \log n)$, so the mapping from planar graph to ZPG garden is polynomial. Given a solution, it can easily be checked in polynomial time, since a solution is never bigger than the garden it solves. This completes the proof that ZPG is NP-complete.

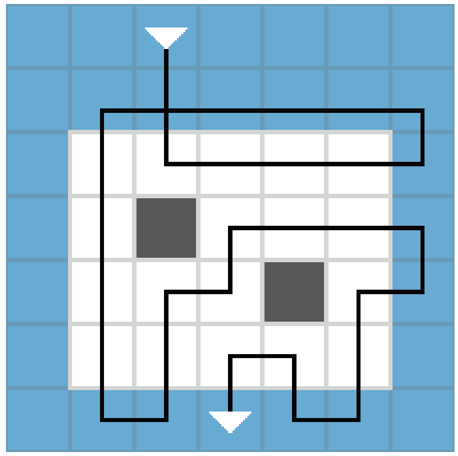

Figure 1: ZPG example garden, with solution. Sand is white, walkable squares are blue, and rocks are grey.

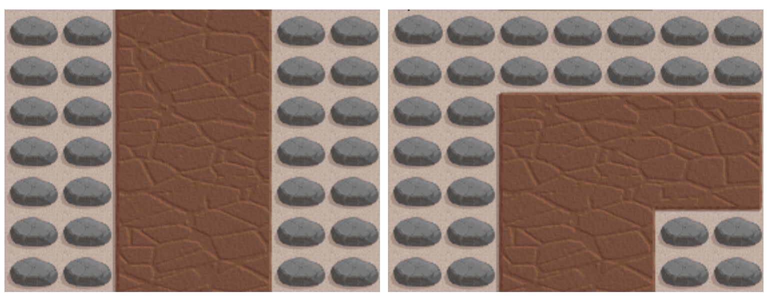

Figure 2: (Left) Straight edge and (Right) corner tiles, represented using actual game graphics. Sand is coloured beige, and walkable squares are brown. Rocks are grey.

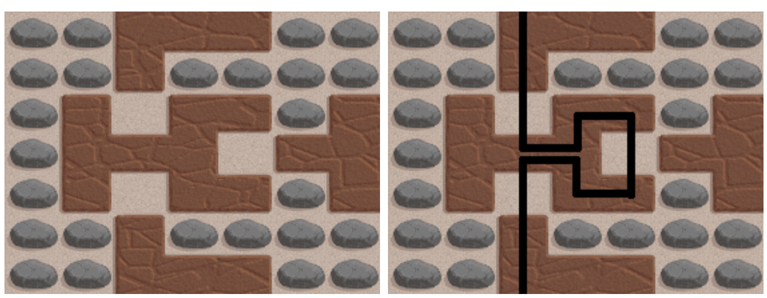

Figure 3: (Left) Node tile, and (Right) example path.

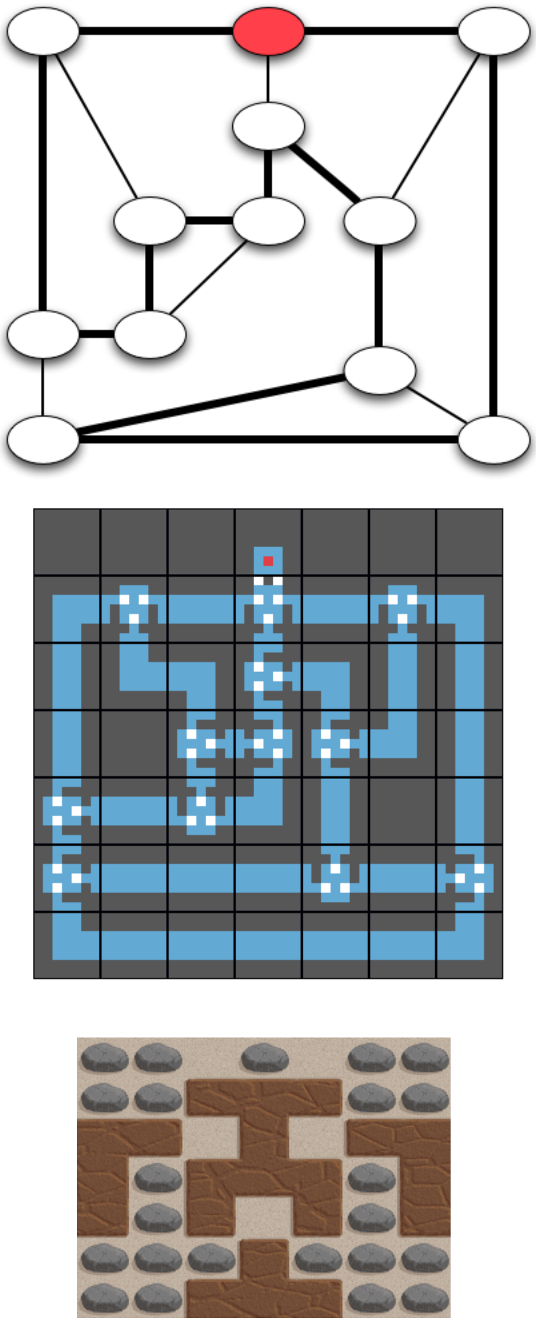

Figure 4: (Top) Cubic planar graph. (Middle) ZPG representation. As before, sand squares are coloured white, walkable squares are blue and rocks are grey. (Bottom) Gateway tile.

## 3. Discussion

Although our proof is straightforward, it confirms the usefulness of the Zen Puzzle Garden game as a test-bed for automated problem solving methods. Historically, games have provided rich inspiration for the design and analysis of algorithms [6, 7, 8, 9], and our hope is that ZPG will offer a novel challenge. To this end, we have made available, in a public repository [10], source code for the A-star and genetic algorithm ZPG solvers described in [2]. This package includes a self-contained ZPG "simulator", which takes a garden description and a series of moves, and produces the final garden state (and which may therefore be used by any future solution method).

As we argue in [2], future work on ZPG might investigate questions such as "Is it possible to automatically generate hard and easy instances of the problem?", as well as considering the notion of an *aesthetically pleasing* solution. In addition to providing a useful test-bed for new solution methods, the problem domain has real practical significance if we consider the problem of mobile robotics, where a self-avoiding path must be chosen whilst also considering possible obstacles and moveable objects.

## References


[1] Lexaloffle Games. Zen Puzzle Garden, trial version downloadable at http://www.lexaloffle.com/zen.htm.

[2] J. Coldridge and Amos, M. Genetic algorithms and the art of Zen. Proc. IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), Nagar, A.K, Thamburaj, R., Li, K. Tang, Z. and Li, R. (Eds.), pp. 1417–1423, 2010.

[3] E. Friedman. Pearl puzzles are NP-complete. Technical Report, Stetson University, 2002.

[4] M.R. Garey, D.S. Johnson and R. E. Tarjan. The planar Hamiltonian circuit problem is NP-complete. *SIAM J. Comput.*, pp. 704–714, 1976.

[5] H. de Fraysseix, J. Pach, R. Pach and R. Pollack. How to draw a planar graph on a grid. *Combinatorica* 10, pp. 41–51, 1990.

[6] E. Demaine. Playing games with algorithms: Algorithmic combinatorial game theory. Proc. Mathematical Foundations of Computer Science 2001, Lecture Notes in Computer Science 2136 pp. 18–33, Springer, 2001.

[7] A. Junghanns and J. Schaeffer. Sokoban: Improving the search with relevance cuts. *Theoretical Computer Science* 252(1-2), pp. 151–175, 2001.

[8] T.-P. Hong, J.-Y. Huang, and W.-Y. Lin. Applying genetic algorithms to game search trees. *Soft Computing* 6(3–4):277–283, 2002.

[9] G. Kendall and K. Spoerer. Scripting the game of Lemmings with a genetic algorithm. Proc. 2004 Congress on Evolutionary Computation, pp. 117–124. IEEE Press, 2004.

[10] J. Coldridge. ZPGsolve A-star/GA package, source code available at http://code.google.com/p/zpgsolve/