University of
St Andrews

Plan and Context Survey

Automated creation of Nanogram puzzle game with Constraint Programming

Author:
Aigerim Yessenbayeva

Supervisor:
Dr. Christopher Jefferson

June 2, 2017

Abstract

There is a wide variety of decision-making problems, which can be solved with Constraint Solvers. Puzzle games is the representation of the combinatorial problems, which is also could be solved with the usage of the constraint programming. Nanogram was chosen as one of possible examples for combinatorial problem, which can be solved with constraint programming.

This paper will describe how all levels of the Nanogram were generated using constraint solver and measuring the difficulty the puzzle game for a human player.

Contents

1 Introduction

2 Objectives

2.1 Primary Objectives

2.2 Secondary Objectives

2.3 Tertiary Objectives

3 Context Survey

- General theory about Constraint Programming[1, 2, 3, 4, 5, 6, 7, 8, 9]
- Manuals and Concepts about Savile Row and Essence Prime [10, 11, 12]
- Related works about puzzle games, their algorithms of solving them and relation to constraint programming [13, 14, 15, 16, 17, 18, 19]

4 Software Engineering Process

5 Ethics

6 Background

7 Design

7.1 Initial Design of a Nanogram solver

7.2 Initial Design of an instance provider for Nanogram

7.3 Enhanced Design of the Nanogram solver

7.4 Enhanced Design of the instance provider for Nanogram

7.5 Design of the game

8 Implementation

8.1 Initial Implementation of a solver

8.2 Initial Implementation of an instance provider for Nanogram

8.3 Enhanced Implementation of the Nanogram solver

8.4 Enhanced Implementation of the instance provider for Nanogram

8.5 Implementation of the game

9 Evaluation

9.1 Evaluating difficulty of levels

9.2 Reducing the solver time

10 Conclusion and Future Work

11 Appendix

12 Plan

Num	Date	Plan
1	Week 00 (MAY 22)	• DOER and Ethics (submission 22 May)
2	Week 01 (MAY 29)	• Determine the puzzle game • Plan and Context Survey (submission 2 June)
3	Week 02 (JUNE 5)	• Revising the Essence Prime • Start modelling the solver of the Nanogram in Essence Prime Make first column and row Use the pre-made instances for testing
4	Week 03 (JUNE 12)	• Modelling a simple solver • Modelling a simple instance provider
5	Week 1 (JUNE 19)	• Enhancing a solver • Enhancing an instance provider • Interim report 1 (submission 23 June)
6	Week 2 (JUNE 26)	• Implementation of the game itself
7	Week 3 (JULY 3)	• Continue implementation and enhancement of the game • Interim report 2 (submission 7 June)

Num	**Date**	**Plan**
8	Week 4 (JULY 10)	• Optionally do the investigation of harder levels of the Nanogram by creation and solving levels of the puzzle faster • Optionally do the comparison of the difficulty of levels between computer and human player.
9	Week 5 (JULY 17)	• Start writing the final report • Enhancing the project • Interim report 3 (submission 21 June)
10	Week 6 (JULY 24)	• Continue writing the final report
11	Week 7 (JULY 31)	• Fixing the possible bugs and problems with the models and the game • Writing the final report and poster • Interim report 4 (submission 4 July)
12	Week 8 (AUGUST 7)	• Proofreading the report and poster • Cleaning the code and commenting some additional parts

References

[1] Rina Detcher. *Lecture Notes in Computer Science*. 1975.

[2] Rina Dechter. *Constraint processing*. Morgan Kaufmann Publishers, 2003.

[3] Krzysztof R. Apt. *Principles of constraint programming*. Cambridge University Press, 2003.

[4] Krzysztof R Apt. Principles of Constraint Programming. *Annals of Physics*, page 420, 2003.

[5] Ian Miguel and Patrick Prosser. What problems will AI solve in future? An old British gameshow can help explain, 2015.

[6] Vincent Park. AN EMPIRICAL STUDY OF DIFFERENT BRANCHING STRATEGIES FOR CONSTRAINT SATISFACTION PROBLEMS. 2004.

[7] Christian Schulte. Principles and Practice of Constraint Programming. 2013.

[8] F Rossi, P Van Beek, and T Walsh. Handbook of Constraint Programming (Foundations of Artificial Intelligence). pages 281–322, 2006.

[9] F Rossi, P Van Beek, and T Walsh Elsevier. Handbook of Constraint Programming.

[10] E Ssence Description, Peter Nightingale, and Andrea Rendl. Description 1.6.4. pages 1–19.

[11] S Avile R O W Manual and Peter Nightingale. Manual 1.6.4. pages 1–9.

[12] Peter Nightingale, Ozgür Akgün, Ian P Gent, Christopher Jefferson, and Ian Miguel. LNCS 8656 - Automatically Improving Constraint Models in Savile Row through Associative-Commutative Common Subexpression Elimination.

[13] Frédéric Dandurand, Q C H V Canada, Denis Cousineau, Jean Jacques Lussier, and Thomas R Shultz. Solving nonogram puzzles by reinforcement learning Nonogram puzzles. pages 1452–1457, 2012.

[14] Kira Goldner, Ryo Kimura, and Ari Schwartz. Nonokenken : Solving Puzzles Using Integer Programs Solving Nonograms using IP Defining the Nonogram. (65).

[15] Christopher Jefferson, Wendy Moncur, and Ke Petrie. Combination: Automated generation of puzzles with constraints. *... of the 2011 ACM Symposium on ...*, pages 907–912, 2011.

[16] Christopher Jefferson, Wendy Moncur, and Karen E Petrie. Combination: Automated Generation of Puzzles with Constraints.

[17] Helmut Simonis. Shikaku as a Constraint Problem. 2009.

[18] Péter Szeredi. LNAI 3010 - Teaching Constraints through Logic Puzzles.

[19] Ramūnas Zavistanavičius. Nonogram solving algorithms analysis and implementation for augmented reality system Nonogram solving algorithms analysis and implementation for augmented reality system. page 54, 2012.