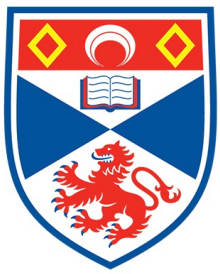

University of
St Andrews

Demo 1 Report

Automated creation of Nanogram puzzle game with Constraint Programming

Author:
Aigerim Yessenbayeva

Supervisor:
Dr. Christopher Jefferson

July 14, 2017

1 Introduction

The main idea of this report is to give the current overview of the project and also evaluate how the initial objectives of the dissertation have been met.

The main purpose of the dissertation project is to implement a solver for Nanogram. Nanogram is also sometimes known as Picross or Japanese Crossword. The solver should be implemented following the concepts of the Constraint Programming using Essence Prime language. The generator of game instances and the analyzer of the game levels should be implemented in Python and passed to the Savile Row to run the solver.

2 Objectives

2.1 Primary

(a) The main objective of project was to implement the generator of the Nanogram, and it was solved using the Python 2.7.13. The generator of the game provides instances of the game, which could be solved by the solver that was implemented in Essence Prime using the constraint programming concepts.

(b) Another thing of the primary objective was to write a system which can generate random instances of the Nanogram and the program is able to execute this thing. But for now it does not guarantee the instance with only one solution. But it never has the cases when it has no solutions at all, since the Savile Row solves all the cases which were tested with the maximum size of grid 30*25 and the grid which were generated randomly with the maximum size of 25*25. However, sometimes it takes more than 20 seconds, which is probably are not solvable by the human at all. These slow executions could be considered as the unsolvable instances.

2.2 Secondary

(a) The secondary objectives are related to the primary ones and could be considered as an extension of it. Since the random generation of the instances could cause unsolvable instances or instances with more than one solution. Therefore, the objective here is to find instances which could have only one solution. This problem is still under the implementation since the implemented random generator can solve the cases when the instance has more than one solution and change it till that point it will have one solution, but only in the cases when the instance had around 2 or 3 solutions in overall. Once the number of solutions is more than 3, then the generator cannot produce the instance with only one solution.

(b) This objective requires to implement Nanogram itself, so that it could be played by the real players. The game must be implemented based on the previous work using UI. So far this part still was not started.

2.3 Tertiary (Optional)

(a) This objective was about investigation of the generation of the harder levels of the game and both create and solve them. So far it was tested with different levels of the game, and some of them were taking a significant amount of time and still need to be investigated more. Moreover it will be necessary to make some statistical analysis finding the correlation of the size of the board and the complexity of the image. If the solver works slower then the implementation of the solver using constraints might need to be reconsidered and possibly changed so that it could work faster.

(b) The last optional objective is to compare the rates of the game levels estimated by the computer versus human.

3 Plan

1. Make analysis based on the random implementation of the instances and through all unsolvable cases, which are actually takes more than 20 seconds or has unnormal amount of solutions (i.e. 100 or more).

2. Try to test the cases which has tremendous amount of solutions and change them by hand in order to check whether they could have one solution if some of the pixels (i.e. cells of the solution board) will be changed.

3. Make a UI of the game and try to compare the rates of the hardness estimated by computers and humans. For that it will be probably needed to involve the participants to play the game.

4 Conclusion

In overall the implementation of the solver in Essence Prime is quite good, since it is able to solve the game with the size of the image field - 30*25 cells. The generator of the instances can also produce solvable instances based on the random chance and based on the images which could be drawn by the human on the white-black image that could be read by the program and produce the instance for the solver. Even if the image would have more than one solution, it is needed to change the image so that it could produce one solution with no losing the meaning of the image. However the program should still be improved by the producing the instances which would be solvable by the computer and human with one solution and through the cases when it is unsolvable due to several reasons which still need to be investigated.