Generating Sokoban Levels that are Interesting to Play using Simulation

Simon Karman (ICA-5521904)
Master Thesis - Presentation
Table of Contents

• Introduction
• Foundational Work
• Improvements
• Experiment
• Conclusion
• **Introduction**
  
  • Procedural Content Generation for Games (PCG-G)
  
  • World Generation
  
  • Research Question
  
  • Foundational Work
  • Improvements
  • Experiment
  • Conclusion
Procedural Content Generation for Games (PCG-G)

- The act of generating content for games using a procedure.

Motivation

- Assisting Artists
- Infinite Worlds
- Many more...
World Generation

- **Mission**\(^{[1]}\) and Space
- Puzzle Generation
- Sokoban
  - Boxes, Goals and Walls
- Large Search Space
- Hard to Solve

---

\(^{[1]}\) Adventures in level design: generating missions and spaces for action adventure games - Dormans, Joris (2010) - Proceedings of the 2010 workshop on procedural content generation in games
Research Question

Can simulation be used to generate Sokoban puzzles that are interesting to play?
• **Introduction**

• **Foundational Work**[^2,^3]

• **Outline**
  
  • Sokoban as a Tree
  
  • Estimate Difficulty Function (EDF)
  
  • Monte Carlo Tree Search (MCTS)

• **Main Limitation**

• **Improvements**

• **Experiment**

• **Conclusion**


[^3]: Data-driven Sokoban puzzle generation with Monte Carlo Tree Search - Bilal Kartal, Nick Sohre, and Stephen Guy (2016) - In: Twelfth Annual AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE).
Outline

- Generation process mapped to a hierarchical non-cyclic tree
- Traversing this tree using a Monte Carlo Tree Search (MCTS)
- Estimate Difficulty Function of Sokoban puzzles using Metrics
Sokoban as a Tree

- Hierarchical and Non-Cyclic
- Branches
- Nodes
  - Root
- Alterations
- Leaf
Alterations
(Detect & Perform)

- Delete
- Place
- Freeze
- Move
- Evaluate
• **Estimate Difficulty Function (EDF)**

• Metrics
  
  • Box

• Terrain

• Congestion

---

Faculty of Science  |  Generating Sokoban Levels that are Interesting to Play using Simulation  

28 June 2018
Monte Carlo Tree Search (MCTS)

- Sokoban as a Tree + Estimate Difficulty Function
- Iteration-based
Main Limitation of the Foundational Work

- We can generate Solvable Challenging Sokoban Puzzles, however:
• Introduction
• Foundational Work

• **Improvements**
  • Alteration Extending
  • Symmetry Reduction

• Experiment
• Conclusion
Alteration Extending

- Impact of Move alterations
- Use a dedicated search algorithm

Push Alteration

- Flood-fill reachable tiles
- Find possible box pushes
Symmetry Reduction

- Reduce symmetry from search space
- Find unique puzzle configurations
- Hard-coded initial five layers
- 2454 -> 12 (reduction of \(\approx 99.5\%\))
Improvements – Symmetry Reduction

Score vs Iteration
average of 10 puzzles with 500k iterations per puzzles

- Foundational Work
- Push Alteration
- Push Alteration + Hard-coded Symmetry Reduction

Number of Iterations
• Introduction
• Foundational Work
• Improvements

• Experiment
  • Setup
  • Output
  • Correlation

• Conclusion
Setup

• Hypothesis

  • The puzzles from the **improved** work, on average, are more interesting than the puzzles from the **foundational** work

User Study

• 4 Datasets: **Foundational Work** vs **Improved Work**
• Number of Iterations: 100,000
  500,000
• Sample Size: 40 users

Tool

• Automatic-Gathered Data
• Question-Gathered Data
Output

- 100.000 iterations: **75%** of the users find puzzles from the improved work more interestingness and on average rate the puzzles with 0.5 stars higher.

- 500.000 iterations: **60%** of the users find puzzles from the improved work more interestingness and on average rate the puzzles with 0.2 stars higher.

- More playtime (x1.9)
- More moves to solve the puzzles (x1.75)
- More overall moves (x1.8)
- More box pushes (x1.4)
Correlation

- Comparison between variables from the output
- 0 -> Weakly correlated
- 1 -> Strongly correlated

<table>
<thead>
<tr>
<th></th>
<th>Playtime</th>
<th>Quickest Solve</th>
<th>Number Of Moves</th>
<th>Number Of Box Pushes</th>
<th>Number Of Resets</th>
<th>Perceived Interestingness</th>
<th>Perceived Difficulty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Playtime</td>
<td>1,00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quickest Solve</td>
<td>0,40</td>
<td>1,00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number Of Moves</td>
<td>0,67</td>
<td>0,84</td>
<td>1,00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number Of Box Pushes</td>
<td>0,71</td>
<td>0,59</td>
<td>0,87</td>
<td>1,00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number Of Resets</td>
<td>0,62</td>
<td>0,42</td>
<td>0,63</td>
<td>0,72</td>
<td>1,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Perceived Interestingness</td>
<td>0,17</td>
<td>0,25</td>
<td>0,28</td>
<td>0,28</td>
<td>0,15</td>
<td>1,00</td>
<td></td>
</tr>
<tr>
<td>Perceived Difficulty</td>
<td>0,39</td>
<td>0,32</td>
<td>0,40</td>
<td>0,41</td>
<td>0,40</td>
<td>0,57</td>
<td>1,00</td>
</tr>
</tbody>
</table>
• Introduction
• Foundational Work
• Improvements
• Experiment

• Conclusion
Conclusion

Can simulation be used to generate Sokoban puzzles that are interesting to play?

No

- Improvements
- Lack of key moves
- Simple Metrics can not capture the underlying concepts of interestingness
Questions?