Final Project Proposal

CS 5100, Fall 2016

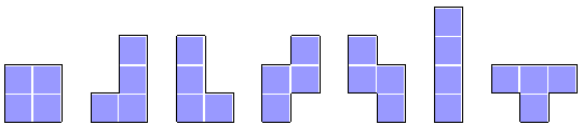
PROJECT TITLE:
SmarTetris

TEAM MEMBERS
Xiao Wang, Ling Zhang, Xuliang Qin

PROBLEM STATEMENT:
The aim of this project is to build an AI agent for Tetris game using search, reinforcement learning and genetic algorithms. The automatic agent is expected to play better than humans in acceptable time. This project also tries to extend the original rules of Tetris and solve the more challenging problem.

MOTIVATION:
Since invented in 1984, Tetris has been popular among all ages. While it's a good entertainment, it's also intriguing for us to look into it in a more serious way. As it's proven that finding the optimal solution for the offline-version Tetris is NP-complete, we are interested in applying the AI techniques we learnt to Tetris to get a good policy in acceptable time.

BACKGROUND:
The Tetris game is played with pieces as shown below [5]:



According to *Tetris is Hard, Even to Approximate* [2], the "offline" TETRIS problem as is describe precisely as follows:
Given an initial game board and a finite sequence of Tetris pieces.
Question Can the Tetris pieces, supplied in the order given, be translated and rotated such that the game board will be cleared with these pieces, the last piece of the sequence filling the final gap?

PLAN OF ACTION:
1. Analyze the problem, research the rules in detail
2. Data design: states, features, actions, rewards, discount
3. Infrastructure: build a game engine to generate blocks, calculate scores, count time and visualize
4. Try and implement different algorithms:
   Search (challenges: time complexity)
   Reinforcement learning (challenges: offline issue, training time)
   Genetic algorithm (challenges: local optima problem, randomness)
5. Evaluate the algorithms based on scores, survival time and efficiency
6. Compare the algorithms, improve them if possible
7. Extend the rules of original Tetris to form a more challenging problem and repeat step1-6

EVALUATION:
- Evaluation factor: total scores, survival time, computational time
- Baseline: place blocks at the lowest position every time
- Oracle: manually play the game
- Minimum requirement: better than baseline
- Qualified criterion: better than oracle
- Demonstration: visualization of the evaluation factors and automatic gameplay

RESPONSIBILITY:

Infrastructure: Xiao Wang, Ling Zhang
Algorithm: All
Evaluation: Xuliang Qin, Ling Zhang
Paper: All

REFERENCE:

[1] Istv´an Szita. Reinforcement learning in games. In *Reinforcement Learning*, pages 540–577. Springer, 2012.
[2] Erik D. Demaine, Susan Hohenberger and David Liben-Nowell. Tetris is Hard, Even to Approximate. In *Computational Geometry*, pages 20–112. Springer, 2003.
[3] Tetris, Inc. http://www.tetris.com.
[4] Istv´an Szita and Andr´as L¨orincz. Learning tetris using the noisy cross-entropy method. Neural computation,18(12):2936–2941, 2006.
[5] R. Breukelaar, H. J. Hoogeboom, and W. A. Kosters. Tetris is hard, made easy. Technical report, Leiden Institute of Advanced Computer Science, 2003.
[6] S. J. Russell & P. Norvig. Artificial Intelligence - A Modern Approach. In *Part V Learning,* pages 566-645.