AI Project Proposal

Ethan DiPilato, Brett Grossman and Abriana Nash

The Problem

The problem is to take an already working source code for the game of Tetris and implement an AI to play the game using a few different algorithms. We will be using a random approach as a baseline, and create greedy and optimal approaches that we will compare and contrast. Tetris is a game which is prime for playing under a search algorithm approach, as the game has mathematical strategies that can be applied.

Background

Tetris is a game played on a grid of size 10 cells wide by 20 cells high. The game consists of various shaped block being presented to you that all fall towards the bottom of the grid at a standard movement rate. As the gameplay continues the movement rate will speed up at certain intervals and certain rate increases. The user's goal is to arrange pieces along the bottom of the grid in a manner that keeps them from losing. Loss occurs when the arrangement of pieces surpasses the height of the grid. The user can keep this from happening by arranging the blocks to clear lines. Lines are cleared when a horizontal line on the grid is completely filled. The user is presented with one advantage in that they have a preview of the next piece that they will be given. The user is also typically allowed to drop the current piece they are manipulating at a faster rate.

Tasks/Schedule

- Choose source code to integrate AI with (by June 23)
- Be able to retrieve problem from Tetris source code (by July 13)
- explore different Tetris strategies to design greedy and optimal approach (by June 26)
- Write random search agent (by July 13)
- Write the greedy and optimal search agents (by end of July)
- Interface AI with Tetris source, i.e. AI needs a way to give piece manipulation instructions (by July 13)
    - receives instruction input from given search algorithm
    - outputs manipulation instructions to program in place of user keyboard input
- Perform tests and record results of various approaches for comparison and presentation (do during last week of project)
- Compile data into a presentation/packet with some conclusion on compare/contrast between greedy and optimal approach (do during last week of project)

Work Assignments

Ethan – find source code, explore game strategies, retrieve problem from game, data collection and presentation/packet creation

Brett - explore game strategies, write search agents, data collection and presentation/packet creation

Abriana - explore game strategies, write interface, data collection and presentation/packet creation

* all collaborate on agents, interface, problem as needed