TCSS 435 - Artificial Intelligence
Assignment 3 - 2048

Guidelines

This assignment consists of programming work. Solutions should be a complete working JavaScript program including your original work or **cited contributions** from other sources. These files should be compressed in a .zip file for submission through the Canvas link.

This assignment is to be completed on your own or in a group of two or three. If you choose to work in a group this must be clear in your submission. Please see the course syllabus or the course instructor for clarification on what is acceptable and unacceptable academic behavior regarding collaboration outside of a group of two or three.

Assignment

The popular web game 2048 (designed by Gabriele Cirulli) is a game in which a single player plays against the randomness of the game. Your task is to create an AI that can play the game well enough to reliably win. If your AI is winning then try to optimize it to get as high a score as possible.

You are provided with a large code base including Gabriele Cirulli's original game simulator with additions made by Chris Marriott to allow for artificial agent control. A sample agent is included that has a very simple strategy. It selects the first move in the list [UP, LEFT, RIGHT, DOWN] that is a legal move (results in a new state). The agent uses a clone of the gameboard to simulate the results of future moves. You must replace its selectMove method with your own strategy. I recommend using Expectimax or Monte Carlo simulation. In both cases you will have to also rely on an evaluation function of the current game state.

Your agent will be evaluated by allowing it to play as many rounds of 2048 it can finish in 10 minutes. There is a trade off between choosing moves quickly versus spending time selecting the best move. All agents capable of winning the game will be submitted to a friendly in-class competition for highest score.

Specifications

In agent.js you will find an AgentBrain class and an Agent class. The AgentBrain is a clone of the 2048 game board that will allow you to advance the game by steps and add your own tiles in order to evaluate possible moves. To use it you will want to first clone the current game board. I do this like this:

- var brain = new AgentBrain(gameManager);

where gameManager is current game board. You can also clone an agent brain in the same way (by passing it into the constructor of AgentBrain).

Once you have the cloned brain you can advance moves using the move method:

- brain.move(direction)

where direction is an integer using the following map:

0. Up
1. Right
2. Down
3. Left

You may wish to comment out line 85 in the agent brain. This line adds a random tile after a move. You probably will want to control where the tile is placed when using expectiminimax so you should replace this with a function that adds a specific tile to a location of your choosing.

Otherwise all your modifications should be to the Agent class. I have provided a simple agent as an example that selects the first available move in the order above.

Submission

The following files are provided for you:

- 2048 code base - html, css, and js files for the 2048 game
- agent.js - includes a simple agent and an agent brain
- agentManager.js - interfaces the agent with the 2048 game

You will submit only:

- agent.js - your agent code