Tetris Analysis

Isabella Zhu

Question 1

It took five generations to get to 50,000 points. However, I ran my code until the seventh generation, in which there were two strategies over 50,000 points. It took about ten hours to get to seven generations.

Question 2

Best Strategy: (0.4801554305084115, -2.242534378609291, 0.15438167696416305, -0.4091602643405545, 0.3141809641978417, -0.5117290627239632, 0.6040595464656984, -0.23652468374239244)
This supposedly scores 60,604 points on average. However, when I ran play_game on this strategy, I got points ranging from the low 1000s to 150k.
In order, the coefficients measure

1. highest column height
2. number of "holes"
3. the number of removed lines
4. difference between largest and smallest column heights
5. maximum "well" depth
6. sum of all "well" depths
7. number of blocks
8. weighted sum of blocks (closer to top means higher weight)

I did not expect there to be any positive coefficients except for number 4. It's strange that the highest column height, max well depth, and number of blocks are weighted positively. The number of blocks may be weighted positively because the weighted sum of blocks is weighted negatively, so they "cancel" out.

Question 3

This assignment was fine, but not my favorite. Here are the things I didn't like about it:

1. Run time is too long. It's true that this could be reduced if I optimized my code more, but even with optimized code it would take at least 20 minutes. Also, this is completely my fault, but I felt very frustrated when I ran my program for 8 hours and realized that I forgot to add the code to save my generation, so I had to rerun it.
2. The fitness function has a lot of randomness in it. Playing a game has a lot of randomness, based on the sequence of randomly generated pieces. Although there is significant improvement in the first three or four generations, it appears that any improvement after that in the "best" strategy is because of luck. When I tried my top strategies from later generations, they had a lot of variance and would not consistently reach 50,000.

I think you should keep this assignment, but maybe set a lower score to reach. 20,000 seems like a good standard, because students can’t just guess coefficients to reach that score and it is high enough to get the concept of genetic algorithms across, but not too large to take forever to run.