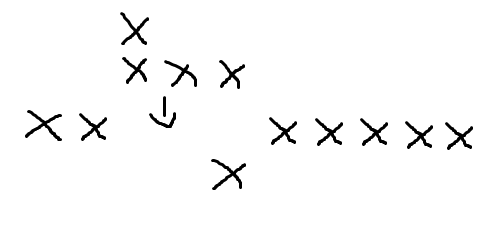
1. My generation 0 had ones that had over 50k. Because of that though, generation 0 takes a while to run, with a total of 1107 seconds or 18 minutes on my school computer. At first, my code had a lot of different variables. I eventually removed the standard deviation of the heights (that made the code a lot slower), number of wells, deepest well, and total well depth, a closeness in height to the middle of the board, max height, total holes depth, number of rows with holes, and if the board just got a tetris. Then, I realized that there was an issue with my modeling, which caused my code to be a lot worse. When changing the max height of each column variable, I just made the ones that needed to go down go down by one. However, sometimes it can go down more than one. In this picture, some go down by two and some by 1. After fixing that, my code became a lot better.
2. Best: (-1.9508615680402759, -0.48043918373960137, 0.9177545348105238, -1.0643673500359405, -1.7842614241761972) Score: 131044.0
   1. They measure flatness (total difference between columns), total height, lines cleared, holes, and row transitions (number of transitions from blank to # and visa versa) respectively
   2. Most of these are pretty expected. I thought that total height would be bad, lines cleared would be good, holes would be bad, and row transitions would be bad. However, I didn’t think that flatness would be that bad. I thought that too flat wasn't good, but somewhat flat was good too. However, it seems like the flatter it is, the worse it is.
   3. Another thing I noticed was that the lines cleared for my top ten were pretty random. The best one was highish, but some were closer to 0. Some other ones were even pretty negative. Everything else was relatively consistent throughout.
3. This assignment was pretty fun but also pretty frustrating. It was pretty annoying trying to find what exactly went wrong with my code, which caused me to spend a lot of time debugging it. I think you should keep this lab, but one thing I would suggest is to make the tetris modeling lab have more cases, such as placing multiple pieces. This would make sure that the tetris modeling has no errors in it, saving a lot of headache later.