Jennifer Frase

Requirements:

- Make a tournament based on assignment 3
    - Allow the user to pick how many creatures are on the teams
    - Allow the user to pick what creature plays in each round
    - Allow the user to name each creature
    - Need to allow the creatures to score points
    - Need to add strength to winning creature
    - Need to report the top three scoring creatures
    - Need to report the top team

Design:

Mostly the same as last week:

For the creatures: in order to have points and a team and add strength, need to add integer variables for points, maxHealth (so that adding health doesn't go over max), team. Need to write simple get functions for those variables as well as simple add functions for the points and strength.

For the fight function, need to add the points allocation to the players and add the loser to the map

After the fight, make the winner get points where it is losers ranking times 10 divided by winners ranking plus the amount of damage sustained by the winner

For the lineups – use two queues, one for each lineup. For the rankings, use a map where the key is their scores and the value is the creature.

Use nested for loops to initialize the lineups, and push the creature into the queue at the end of the internal loop

For the tournament, repeatedly send the first creature in each queue to fight as long as neither queue is empty.

After the fight, add the winner back into their queue.

Add back strength based on the number of creatures still in the queue

Add 1 point for every two creatures in the queue

Look at the returned creature's team and add to the appropriate queue

Once a team is empty, move all the remaining creatures in the other team over to the ranking map.

Loop by Looking to see that the creatures score is not already in the map

If it is add 1

Add the creature to the map with it's points as the key

Once all the creatures are in the map, loop through them to get their points- start at the end and work forward

Print the first(last in the map, first in the loop) three creatures with their score and team

Add the creature's points to their team's points

Print whichever team had the highest number of points.

Testing

See testing doc.

Reflections

For the scoring, I calculated the average percentage that the creature is likely to win against all creatures using my test data from last week. This gave me their ranking score. I then figured that in order for the stronger creatures to get less points against a weak creature I could have the losing ranking be divided by the winning ranking. However, this had the drawback of similar ranking opponents not getting very much, even if they are strong. I then thought that I could also include how much damage the winning create takes as a factor to their scoring. I figured that if the creature was against a weak opponent then they would get much damage, but if they were against a strong opponent then they would probably sustain a great deal of damage. If they were against similar ranking opponents, then the likelihood they would sustain damage would be high and thus, they could get some points.

For the strength recovery, I most though along the line of fighting. When you are in a tournament, you get to rest while you wait your turn. The longer you wait, the most strength you would recover. As a result, I made the recovery amount dependent on how many creatures were still on their queue. In this way, if the creature waits a long time, they can recover a lot, but if they have to fight immediately, then they don't recover at all. However, I thought that since this is based on resting, then the strength shouldn't recover too quickly, so I made it so that 1 strength is recovered for every two creatures ahead of it.

I thought testing was a little weird this week, primarily because the prompt says that the user should only wait for the results after entering the lineup. This meant that they don't see any of the

internal numbers that calculate those result. If you have your program do this, then how are you supposed to test that the results are accurate while showing that in the testing section? The way I tested it was to have code that showed the numbers and then I commented it that out as I went along making sure each part worked.