**The Robot Fight 1.0 Data Project**

Using the BattleTech game JSON files of mech definitions as a base, I will be reverse engineering my guess as to the battle system in BattleTech by observation and then adding things I think fun. My hope is to then build some data science into stored battle data and come up with reports as if I were wanting to study these robots. Future ideas include an entrepreneur building a robot battle coliseum and managing all aspects of fights, adding pilots to the mix, adding a manufacturer outlet, allowing customization of robots, upgrading robots, and other ideas inspired by BattleTech and other games. I also seek input and ideas from other people to make this an ongoing and growing portfolio of understanding for use in networking efforts and demonstrating skills in interviews. For now, I will be focusing on the following ideas for version 1.0, marking some as ideas to implement in 2.0 or beyond. For #1 and #2 below, I will just be importing the JSON files from BattleTech and saving them in a Mech Class.

1. Robots have armor on various parts of the body
   1. Head and left, right, and center torso, arms, and legs
   2. Some parts have rear armor
2. Robots hold an inventory on those body parts that can contain:
   1. Weapons
      1. Deal Damage to Armor
      2. Have minimum, maximum, and optimal ranges
      3. Have one or more shots per firing
      4. Some weapons have ammo
   2. Equipment
      1. Jump Jets
      2. Heat sinks
      3. Other
3. Robots Take Damage accounting for:
   1. Weapon damage
   2. Weapon Default Accuracy
   3. Range Percent Modifier to accuracy
   4. What part of the body gets hit, based on percentage
   5. Other modifiers, like Pilot and Terrain
4. Storing battle data:
   1. JSON files
   2. SQL Server database
   3. Generate random samples of data to the millions of records
5. Data Science on battle data:
   1. Get input and ideas from others on what to ask from the data
   2. Get Data Science in Python book or use Google and learn to make stuff
   3. Compare performance between Python and SQL when data gets large

Features to add to 1.0 (as of 8/15/2020)

1. Storing Battle Data in CSV, JSON, and/or SQL Server
   1. Mech current status each round or turn
2. Testing for Bad Data
3. Create Basic Reports
4. Data Science Questions
   1. How should I fight?
      1. Fire everything as much as possible, based on current design
   2. How should I configure my robot?
      1. Generate random configurations of mechs
      2. Test against battle data
   3. What percentage of fights will I win?
      1. Random choices as currently programmed
      2. Max alpha strikes for each opponent

Features for 2.0

1. Heat Management
2. Dynamic Percent to hit body parts
3. Default accuracy percent using constant
4. Random coin flip to determine order
5. Choosing to do nothing or defend
6. Toggle Choosing Weapons
7. Reflect Heat Generated when using weapons when toggling
8. Show Destroyed weapons on the Attack Selection of Weapons with an X
9. Stability Damage
10. What is displayed when toggling
11. Multi Attack
12. Constants and adjusting accuracy
13. Stray Shots
14. Critical Hits
15. PyGame UI
16. Using Rear Armor
17. Display Current Status of both robots
18. Weapon Ranges
19. Other Equipment
20. Pilots
21. Terrain and Environments
22. Angles and Mapping
23. Upgrading Max Armor
24. Salvage
25. Hardpoints
26. Destroying Ammo boxes when attacking but a body part isn’t destroyed
27. Melee attacks