MIDS – INFO W18 – Python Bridge Course  
Solo Project  
Jason Becker

F1 Fantasy League

Objective:

Create an object oriented module that facilitates the management of a fantasy F1 league in Python.

Background:

Formula 1 is an international racing competition consisting of 22 drivers competing in 21 races over the 2016 season. A fantasy league is a competition between fans in which they draft racers and accumulate points depending on the performance of their team of drivers. A commissioner currently has very few commercially available options to manage the tracking of a fantasy Formula One league. This project aims to allow a commissioner to easily create teams for each player in the league, track the results of a race, and track scores (based on various agreed upon criteria).

Implementation:

The first object would be a driver (let’s use Danlil Kvyat as an example). They will have attributes like a team (Red Bull), and results (Qualified in 4th for the America Grand Prix), and driver points (currently 0 because the season hasn’t started.

The second object type would be a race. Race results would have a method to pull in final results after a race is complete by either scraping Wikipedia or one of the several websites that are available. The attributes would store the qualifying position, grid position, finishing position, and any grid penalties accrued. When this information is retrieved, it would need to push these results to the update driver attributes and award them with driver points. The more points a driver earns, the more they cost to draft on your team. Every new result would update the calculation of their cost for subsequent races as well.

After an import, it is possible that an individual result needs to be updated. For example, the by-laws state that the starting position is irrespective of any gaps, even those created during the formation lap. Some data sources won’t take this into account and as a result award movement points to a driver for increasing their position without overtaking. Changing the starting position of two drivers would need to push updates to the race results, both drivers, and any players who selected those drivers for that week. Changes need to be error checked as well to make sure that no two drivers finished in the same position.

The third object type would be a player. A player object must be able to add drivers to their team, and drop drivers when you realize that they’re absolutely terrible. Every has a player budget, and those changes must not exceed the budget. Bonus points are awarded for using a team that is below the budget as well. When race results are pulled, the player will need an updated score for that week.

Since this is a fantasy sport, it thrives on trash talking and being overly competitive: each object will need methods for displaying the current standings. Driver standings, player standings, summary tables of a race, etc. will need to be available. If time allows, a quick command to create a line graph with matplotlib would be a great method as well.

Extensions:

Although it is outside the scope of this current project, a framework like this could be used to build a series of dashboards for live fantasy results. A group of viewers can watch the race on one screen while another screen shows live scores using the timestamped XML results.

Appendix A: Summary Of Scoring Rules

Each team must consist of no more than 5 drivers, totaling no more than 33 points. A team using less than 33 points will receive a bonus equal to the number of points that it is under.

A driver’s cost will be recalculated every race, equaling the sum of the drivers points in the previous four races, divided by the sum of all points earned by all racers in the previous four races, multiplied by 100, rounded to the nearest whole number. The first race (Australian Grand Prix) will not be costed or scored.

Drivers earn 1 point for reaching Q2, and additional point for reaching Q3, and one bonus point is awarded for pole position. Ten points are awarded for first place, Nine points for second place, etc. One point is awarded for finishing ahead of your teammate in a race. Three points are awarded for classifying (90% of a race completed, or one point for completing at least 50% of a race. The driver who clocks the fastest lap is awarded two bonus points. Drivers may earn movement points equal to their grid position minus their finish position. Movement points cannot be negative, and are only awarded to a driver that finishes in the points (top ten).