Josh Antonson

Jantonso

Section G

Choosing the Best Player Available

As I said in the high level design, my program is a settings customizable fantasy baseball draft, as well as automatically drafting for one of the teams, it drafts the best possible player available depending on a variety of settings. The major determinant of the auto-drafting element is whether or not the league is using “Points” or “Rotisserie” as its scoring setting. If the setting is “points”, then the user has assigned a point value to each setting, i.e 8 for “HR”, 6 for “Wins”. If the setting is “Rotisserie” then the user selects which settings will be used and each team the sum of each stat to the other teams to determine the winner, for example if my team has 10 HR’s and another team has 5 HR’s then my team would win that category. My program first gets a list of all the players available and their projected stats from a website. If “Points” is chosen it will then determine the total projected points for each player based on the values assigned to each stat, for example if a player was projected to hit 10 HR’s and a HR is worth 5, then his projected points would be 50 + the projected points of his other stats. If the “Rotisserie” setting is used then it determines the best player available by ranking your team’s totals in each individual stats to the rest of the league to determine your rank in each category, for example if you are last in HR’s then it will value HR’s greatly in determining the best available player. Once it has a list of the top ranked players it will attempt to add them to the team as long as that players position is empty on the current team.

I also implemented certain features for the auto draft, with the main one being the ability to choose whether or not to use a setting called favorite Team or rival Team. If you choose a favorite team, the auto-drafting program will draft a player from your favorite team if it is within the top 10 ranked players, and will not draft a player from your rival team. I enabled this feature because I know for example being a Cubs fan; I would like to have as many Cubs players as possible and would also not draft a Cardinals player.

The way that I determined most of the user’s chosen settings was through the mousePressed function, as I had a list of all the settings and their corresponding check boxes. If the user clicked the mouse within the check box, then I would either toggle the setting on or off and implement any corresponding drop down features or by then enabling the user to type in the values of the points for if they had chosen point settings.

User Interface

I modeled the actual drafting interface off of the ESPN fantasy baseball-drafting interface. The ESPN interface is laid out nicely, but I think it displays almost too much information, so I decided to implement the key features while avoiding the unnecessary amounts of information. The main features are a list of the available players and their projected stats, the ability to view each team’s roster, a list of the previously drafted players, a draft clock, and an automatically updated pick order feature. The updated pick feature scrolls from left to right with the team up next, I chose to implement this at is impossible to have a fantasy baseball draft without knowing who is drafting and when you are up. The ability to look at a team’s roster is possible by clicking on the team pick order at the top, this feature is laid out to allow each person to view their team as well as the other teams quick and cleanly. Originally, I wanted to have a scrollable available players list, but I soon realized that was extremely difficult in tkinter, so I decided to use buttons to switch between previous 25 and next 25, as well as between pitchers and hitters. I also used a similar method with buttons for displaying the previously picked players, with the ability to look at the drafted players 25 per page.

For the settings page I wanted to keep it very clean and just allow the user to quickly look through and choose the settings that they wanted. The majority of the decisions are done using check boxes, which was an easy implementation and is an easy and quick way for the user to see what they have chosen/what they still need to choose. For choosing the values assigned to each category if the user has chosen point settings, I decided let the user assign the values by typing from their keyboard, as having 10 different check boxes would have been unnecessary. I had a dropdown/pop-up feature for when the user is choosing a favorite team and rival team, because I wanted them to have the ability to see a list of all the teams, while not having the large list full of teams appear when they are choosing other settings.