Nick Willertz

Software Engineering

Weekly Reports

Week of 2/17 – 2/23

After some hiccups with deciding on our project, the Yellow team has picked a C# Sports League generator/manager project and started making some good progress on it. We wrote 13 stories and separated the first couple of stories into about seven tasks. On Sunday, 2/17 we finalized what exactly those tasks are, and then on 2/19 we spent some time programming. We pair programmed for about two hours on 2/19 and all contributed a lot to the code. I think we are off to a good start and already have about six or seven crucial project test methods working successfully.

Week of 2/24 – 3/2

We made a lot of progress on the project and generated the basic functionality of the League, Team, and Player classes and how they work together. On 2/24, we completed the testing of many of the basic functions for each of those classes, like creating, adding, and removing leagues, teams and players. The rest of the week was spent on generating an even schedule for a League using its current list of Teams. This algorithm was a challenge and took multiple hours with pencil and paper, but we got it to work with the manipulation of a list of Games.

Week of 3/3 – 3/9

We fixed a problem with the generation of leagues and now the League class takes in the number of teams that make the playoffs instead of max teams in the league parameter. We then started working on a main method in which a user could interact with to create, add, and remove leagues, teams and players. We spent the rest of the week on this story and still have a lot more work to do on it. We ended up having to split this main method story into a few different stories for working with league, team, and player individually and had to take out one additional story from this current iteration.

Week of 3/10 – 3/16

Spring Break

Week of 3/17 – 3/23

We were able to complete the current iteration, which was creation of a basic functioning text-based user interface. A user can create a league, create/add and remove teams from a league, and create/add and remove players from a team. We were able to make this basic functionality of the UI work and handle much of any bad information that the user may enter. The next steps will be taking the UI farther with more features like generating the schedule of a league and entering the results/scores of games in the league. The backend code for these features has been written, but as we learned with this second iteration, building the UI to work with the backend code can be challenging.

Week of 3/24 – 3/30

We started by writing tests for all of the UI features to run without user prompts. Therefore, we rewrote all of the UI functions to take in parameters for the values that the UI prompts the user for in order to test these functions more easily. Then, we worked on the new feature for the UI in which the user can generate a schedule for the league they have created based on the number of weeks they select and can look at that schedule and which teams play every week. We were able to get this feature working with just a little bit of trouble displaying the schedule properly.

Week of 3/31 – 4/6

We continued work on the UI generate and display schedule features and have these features working pretty well now. We then created the feature that allows a user to look at the schedule and update the scores of specific games. The program requests the game that the user wishes to update the score of and prompts them for each team’s score. The program then updates the score of that game and declares a winner. The game class stores the score and updates the records of the individual teams. We also finished the feature for displaying a team’s roster and record. When a game score is updated,, the records of the teams are updated based on whether they won or lost.

Week of 4/7 – 4/13