Joseph Weiner

CMPT 220L-112-17S

Professor Juan Arias

Java-Catch: The First Ever Fully Comprehensive Pokemon Catch-Rate Calculator

The project idea that I have come up with is an application to determine the catch rate for any and all encountered Pokemon in the popular video game franchise of the same name. Many players of the game decide to attempt additional challenges in completing the game that adds to the difficulty of the vanilla meta game. Such challenges like the "Nuzlocke," "Moneylocke" and "Quadlocke" add additional rules and clauses to the already present rules of the game. These challenges have been popularized by the internet communities around the game and usually, a challenge participant will broadcast their whole run through of the game while doing the challenge. The original "Nuzlocke" challenge is essentially a limit on not only how many Pokemon one may catch, but also when and where. A Moneylocke would be the inability to buy any items including the trademark "Pokeball" and of course the others usually explain themselves if not a link can be found here explaining a variety of these challenges.

So knowing more about these types of challenges presents us with our problem, that being the frustration that many players deal with using up their already limited resources on uncalculated encounters. If a player only has as many items as they pick up, their resources are greatly limited and every use counts greatly. What I propose is to create an application that can actually calculate over any number of simulations the probability of catching a Pokemon is.

This would work by essentially creating a java program that will be able to interpret and calculate the already documented algorithm for standard catch-rate. The program will also have to calculate the modifications to the algorithm that are given by each different "ball." The program will also have to apply these calculations to any Pokemon the user inputs, every Pokemon coming with a different catch-rate of their own which will be a constant outside of the algorithm and the conditions and variables which change this algorithm. Finally, the application will need to run the desired simulation as many times as a user wants and then take an average of the success rates. All of this will come packed in a neat GUI, hopefully with images and animated loading bars.