Report Darts 501

The way I choose which part of the dartboard the player should for during simulation is by using if else statements that check the score. In 301 the players throw for inner bull while the score is more then 70 because that gives the most points and is the fastest way to finish. Once the score gets near 50, they throw at targets that would allow them to make it exactly 50. At 50 they throw bull to end game. During 501 I used the information from a checkout chart I found online to decide which target aim for. Each player has a chance of missing the target he was aiming for and hitting one of the 8 segments surrounding his target. Each segment has a 1/8 chance of getting hit.

The user can interact with my program through the menu where they can choose which gamemode they want to play (darts 301, darts 501, darts 501 interactive). The user can also change the inner bull chance in the setting and the outer bull chance, single chance, double chance and treble chance in the advanced settings. In the darts 301 and darts 501 menu the user can choose whether or not they will get stats at end of the game and stats about the different throws being made displayed. The reason I included that option to not display the throws being made during simulation is that I found on that the program runs faster.

One of benefits of object-oriented programming are that since everything is modular it is a lot easier to troubleshoot because a problem in one file does not make the entire solution fail. Another benefit is that you don’t have to write multiple same variables and function every player you can just have them be part of your player class and then reuse them by creating new player objects. It also allows you to focus on separate problems by looking at separate classes. For example, I might want to focus on my players functions and variables, so I do not want to get distracted by the functions used for displaying text. In addition, Object oriented programming is securer because the private variables in each class cannot be accessed or by a another non derived class without using getter or setters.