1. Create an object of player class for Joe and Sid

2. Create an object of dartboard class

3. Ask the user if this is a 1 or 2 player game

3.1 If it's a 2 player game

3.1.1 Ask the user which player goes first

4. Ask the user the number of games that are to be played

5. Repeat games user entered number of times

5.1 reset all player.currentScore to 301 and player.dartsThrown to 0

5.2 Start game loop, repeat until one player.currentScore == 0.

5.2.1 Evaluate the current score and decide a course of action

5.2.1.1 if the currentScore is < 50 and > 0, set currentscore to value prior to the dart thrown up until they finally end up on 50.

5.2.1.2 if the currentScore is > 50 and <= 70, call throw\_Single(int), aiming for the remaining value to get to 50 (currentScore-50).

5.2.1.3 if the currentScore is <= 100 and > 70, call throw\_Single(int), aiming for 20.

5.2.1.4 if the currentScore is > 100, call throw\_for\_bull, aiming to score max points.

5.2.1.4.1 if player hits a bull, add one to player.BullsHit

5.2.1.5 if the currentScore is == 50, throw\_for\_bull aiming to finish.

5.2.1.5.1 if player hits a bull, add one to player.BullsHit

5.2.2 Send player.currentScore to display

5.2.3 Add one to player.throws

5.2.4 Send player.throws to display

5.3 Add one to player.wins for the player who successfully reaches 0 first.

5.4 Append the winning player.dartsThrown to a vector

5.5 If 2 player game

5.5.1 Repeat loop with other player

5.6 if 1 player game

5.6.1 Repeat loop maintaining current player

6.End loop once number of games has been played

7. Output player.wins for all players to view overall winner.

8. Calculate frequency of game length

8.1 count number of occurrences of a given number within vector

8.2 ((number of occurrences / number of games played)\*100) to get frequency of game lengths for that number of darts as a percentage.

9. Send calculated frequency to display

**UML**

