Pseudocode

1. Preparations
   1.1. Create array of two player objects, array of two corresponding score objects, a dartboard object and an object for dart throw decisions
   1.2. Initialise variables (for # total matches, # total sets, # throws per turn, initial game score, variable deciding who starts a match, …)
   1.3. Prototype functions
2. For (match # < # total matches) — loops through all matches
   2.1. For (set # < # total sets) — loops though all sets within a match
      2.1.1. Set starting player for match to Sid
      2.1.2. While (neither player has won more than 2 games) — loop through all games within a set
         2.1.2.1. Initialise player scores to 501
         2.1.2.2. While (neither player has score 0) — play a game
            2.1.2.2.1. Decide who's turn it is aka. *current player* (CP)
            2.1.2.2.2. Get CP's score and store as pre-turn score
            2.1.2.2.3. For (# CP's throws < 3) — let CP throw three times
               2.1.2.2.3.1. Initialise throw as one that is an invalid final throw
               2.1.2.2.3.2. Decide which score to aim for based on CP's score
               2.1.2.2.3.3. Perform actual throw
               2.1.2.2.3.4. Get "Former score of CP" - "Score from actual throw" (*)
                  2.1.2.2.3.4.1. If (*) larger or equal to 2
                     2.1.2.2.3.4.1.1. Update CP's score to (*)
                     2.1.2.2.3.4.1.2. If (# CP's throws are 3)
                        2.1.2.2.3.4.1.2.1. Change who's turn it is (affecting variable in 2.1.1.2.1)
                  2.1.2.2.3.4.2. Else if (*) is 1, negative, or 0 but the finish throw was invalid
                     2.1.2.2.3.4.2.1. Reset CP's score to pre-turn score
                     2.1.2.2.3.4.2.2. Change who's turn it is (affecting variable in 2.1.1.2.1)
                     2.1.2.2.3.4.2.3. Break
                  2.1.2.2.3.4.3. Else (i.e. if (*) is 0 and the finish throw was valid)

2.1.2.2.3.4.3.1. Set CP's score to 0
2.1.2.2.3.4.3.2. Increase CP's # games won
2.1.2.2.3.4.3.3. Change who's turn it is (affecting variable in 2.1.1.2.1)
2.1.2.2.3.4.3.4. Break
END IF
END FOR
END WHILE
END WHILE
2.1.3. Increase # sets won for winner of the game (i.e. the player that reaches 3 first) by 1
2.1.4. Change who's turn it is to start the next set
END FOR
2.2. Store number of sets won for player 1 in array (+)
END FOR

3. Output final stats over 10000 matches to screen
   3.1. Output occurrences of final match scores for both players on screen by doing this:
      3.1.1. In array (+), count occurrences of final match scores for player 1 and store in array (~)
      3.1.2. Calculate final match scores for player 2
      3.1.3. Calculate percentage with which the 14 possible end scores occur
      3.1.4. Output the above to screen
   3.2. Output the most likely outcome to screen by doing this:
      3.2.1. Find result in array (~) that occurs most frequently
      3.2.2. Accordingly, find winning player
      3.2.3. Output most frequent result and winning player to screen (I decided to also output how many matches each player won in absolute terms)
4. Hold open so that output can be seen