Darts 501 Pseudocode:

`Source.cpp main()`

```
seed random

string name
short int accuracy
char menuChoice = returned value from mainMenu function
int matches = 0
short int p1throw = 0
short int p2throw = 0

    do while menuChoice isnt 1
        if menuChoice is 1
            clear console
            break
        else if menuChoice is 2
            print rules to screen
            call mainMenu function again
    do while ends

    clear console

    loop twice

        do while name is empty OR accuracy is less than or equal to zero
            ask user for players name
            take user input

            ask user for players accuracy
            take user input

            clear console

        do while ends

        if i is 0
            assign name and accuracy to a new Player object called player_one
        else if i is 1
            assign name and accuracy to a new Player object called player_two

    print how the player that goes first is decided

    do while p1throw adn p2throw are zero
        set p1 and p2throw to result from throw_bull with respective players
accuracy passed in

        print to console what score each player threw
        sleep for 1 second

        if p1throw is 50 AND greater than p2throw
```

```
            assign player_one and player_two to a new Game object called newGame,
the player on the left hand side goes first, in this case that is player_one
            print to console which player is going first ( player_one )

        else if p2throw is 50 AND greater than p1throw
            assign player_two and player_one to a new Game object called newGame,
the player on the left hand side ( player_two ) goes first
            print to console which player is going first ( player_two )

        else
            set p1throw and p2throw to zero ( to not break out the dowhile loop
)
            clear console
            print to console that no certain player is first and that the program
is trying again
            sleep for 2 seconds
            clear console
    do while ends

    do while matches is less than or equal to zero
        print to console asking how many matches to play
        take user input

        clear console

    do while ends

    loop for amount of matches set by user
        call gamePlay function

    print results by calling matchResults function

    exit
```

`Player.cpp Player()`

```
    set playerName = Joe, default value
    set playerAccuracy to 70, default value
    set score to 501
```

`Player.cpp Player(string name, short int accuracy)`

```
    set playerName to the value of the name parameter passed in
    set playerAccuracy to the value of the accuracy parameter passed in
    set score to 501
```

`Player.cpp MINUSscore(short int newScore)`

```
    take away value of newScore parameter passed in from score variable
```

Player.cpp RESETscore()

```
    set score to 501
```

Player.cpp INCplayerMatchWins()

```
    add one to playerMatchWins variable
```

Player.cpp RESETcurrentSetWins()

```
    set currentSetWins to zero
```

Player.cpp INCcurrentSetWins()

```
    add one to currentSetWins variable
    add one to playerSetWins variable
```

Player.cpp RESETcurrentGameWins()

```
    set currentGameWins to zero
```

Player.cpp INCcurrentGameWins()

```
    add one to currentGameWins variable
    add one to playerGameWins variable
```

Player.cpp RESETgameTurns()

```
    set gameTurns to zero
```

Player.cpp INCcurrentTurn()

```
    add one to currentTurn variable
    add one to gameTurns variable
```

Menu.cpp mainMenu()

```
    string lineTxt
    char menuChoice

    open file mainMenu.txt for reading

    while there are lines left keep looping
        print contents of file to console line by line

    take user input for menuChoice variable

    return menuChoice
```

Menu.cpp resultsScreen()

```
    string lineTxt

    open file Results.txt for reading

    while there are lines left in the file keep looping
        print contents of file ot console line by line
```

Game.cpp Game(Player* player_one, Player* player_two)

```
    set position zero of the players array to player_one
    set position one of the players array to player_two
```

Game.cpp darts_strategy(short int currentScore, short int currentTurn, short int accuracy, int index)

```
    short int random = random value between 21 and 1

    if players current score is > 501
        reset current players score to 501
        set current turn to 0

    if players current score is <= 501 AND > 100
        call throw_treble function, with a target of 20
        return 2, indicating a treble has been thrown
    else if players current score is <= 100 AND >= 49
        initialise short int variable i to zero

        do while current score is NOT equal to value of current position in
array
            if current score is equal to current position in array AND is NOT
equal to 99
                if current turn is 0 AND current value of throwtype array is T
                    throw a treble with value of current position in throwstrat
array as a target alongside accuracy, take away value returned from players
current score
                    return 2, indicating a treble has been thrown
```

```
                else if current turn is 0 AND current value of throwtype array is
S
                    throw a single with value of current position in throstrat
array as a target, take away value returned from players current score
                    return 0, indicating a single has been thrown
                else if current turn is 1 AND current score is 99
                    throw a single with 10 as the target, take away value
returned from players current score
                else uf current turn is 2 AND current score is equal to the value
of current position in throwstrat array
                    throw a double with value of current position in throwstrat
array as a target, take away value returned from players current score
                    return 1, indicating a double has been thrown
                else
                    throw a double with the value of random as a target, take
away value returned from players current score
                    return 1, indicating a double has been thrown
            else
                throw a double with the value of random as a target, take away
value returned from players current score
                return 1, indicating a double has been thronw
        do while ends

    else if current score is < 49 AND > 40
        throw a double with random as a target, take away value returned from
players current score
        return 1, indicating a double has been thrown
    else if current score is <= 40 AND is an even number
        throw a double with HALF the current score as a target, take away value
returned from players currnet score
        return 1, indicating a double has been thrown
    else if current score is < 40 AND is an odd number
        throw a single with random as a target, take away value returned from
players current score
        return 0, indicating a single has been thrown
    else
        throw a double with random as a target, take away value returned from
players current score
        return 1, indicating a double has been thrown
```

Game.cpp gamePlay()

```
    for loop, looping 13 times for max number of sets per match

        if player 1 AND player 2's set wins are NOT equal to 7

            for loop, looping 5 times for max number of games per set

                if player 1 AND player 2's game wins are NOT equal to 3

                    do while either players score are NOT equal to 0
                        call darts501 function to run one round of darts 501 per
loop
                    do while ends

                    if player 1's score == 0
```

```
                        add one to player ones game wins variable
                        reset score for both players
                    else if player 2's score == 0
                        add one to player twos game wins variable
                        reset score for both players
                else
                    if player 1's current game wins == 3
                        add one to player ones set wins variable
                        reset current game wins for both players
                        break from loop
                    else if player 2's current game wins == 3
                        add one to player twos set wins variable
                        reset current game wins for both players
                        break from loop
        else
            if player ones current set wins == 7
                add one to player ones match wins variable
                call outcome storage function with both players current set wins
as parameters
                reset both players current set wins
                break from loop
            else if player twos current set wins == 7
                add one to player twos match wins variable
                call outcome storage function with both players current set wins
as parameters
                reset both players current set wins
                break from loop
    if statement, same as 2 directly above
```

Game.cpp darts501Round()

```
    short int temp_score = 0
    short int new_score = 0
    char throw_type = '4'

    for loop, looping twice, once for each player in the array
        set current players current turn to 0

        for loop, looping 3 times, once for each turn
            if current turn == 0
                set temp_score to players current score

            set throw_type to value returned by darts_strategy function
            set new_score to players score ( after a dart has been thrown )

            if throwtype == 1 (double) AND current score == 0
                this is a valid win so set the players score
                break from current loop
            else if throwtype == 3 AND current score == 0
                valid win, allow setting of players score to 0
            else if current score is <= 1
                set players score to temp score saved at the first throw of this
round of throws
        if current players current score is == 0
            break from current loop
```

Game.cpp outcomeStorage(int setWinsP2, int setWinsP1)

```
    switch statement, taking setWinsP2 as an expression
    7 cases, 0 through 6, each one does the following
        add one to relative position(case 0 is position 0 etc) in array 1 of
ratio
        break from switch

    switch statement, taking setWinsP1 as an expression
    7 cases, 0 through 6. each one does the following
        add one to relative position (case 0 is position 7, continue counting up
) in array 1 of ratio
        break from switch
```

Game.cpp matchResults(int matches)

```
    call resultsScreen function, printing results ascii art to console

    print to console some simple formatting for player names and frequency
header

    for loop, looping 14 times, length of ratio array

        if value of ratio at the current position is NOT equal to zero
            if i is < 7
                calculate frequency current outcome occurred and add into
position i of ratio percent array
                print ratio as output with some formatting
                print frequency as output with some formatting
            else if i >= 7
                same as statement above, but formatting for ratio is switched
```