**Craps Game**

**Algorithm:**

Separate method to roll two dice and return the sum of the results

* Important to “roll” two separate 6-sided dice as opposed to a random number between 1-12 because the odds of getting a 7 in each scenario is different (1/12 on 1 1-12 roll vs. 1/6 for 2 1-6 rolls)

Roll the dice for the user, and if the user doesn’t immediately get craps (2, 3, 12) or a natural (7, 11), “roll” the dice for 2 or 3 other “computer players” and check the same for them.

All players standing get assigned their “point” as the value they initially rolled, each player continues to roll until they get their “point” or craps

Create array as a sort of “leaderboard” where the player can see who lost first to last

Display leaderboard to user after there is only one person still rolling

**Method Signatures:**

CrapsGame Class:

main(String[] args) – main method  
startGame(Player[] crapsPlayers) – runs the entire craps game, calls other methods  
putFirstAvaiable(Player crapsPlayer) – puts the player into the first available leaderboard spot  
putLastAvailable(Player crapsPlayer) – puts the player into the last available leaderboard spot  
printLeaderboard() – prints the entire leaderboard of players  
arrayContains(Player[] playerList, Player crapsPlayer) – checks if a player is in the array  
getYesOrNo() – gets a simple yes or no response from the user  
getSumOfDice() – rolls two independent dice and returns the sum of their rolls  
checkRoll(int roll) – returns 1 if roll is natural, 0 if its nothing, -1 if its craps  
getUserName() – asks the user for their name and returns the result

---

Player Class:

Player(String playerName) – constructor for the player  
getPoint() – returns the user’s point  
getName() – returns the user’s name  
setpoint() – sets the user’s point  
setName() – sets the user’s name

**Module Structure Chart:**

Diagram

Description automatically generated

**UML Diagram:**

Diagram

Description automatically generated with medium confidence

**Test Set/Plan**:

To help debug, the program will have println statements that print the initial rolls on the dice and check that everything adds up properly

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Roll 1** | **Roll 2** | **Point** | **“turn”** | **Result** |
| Adds to 2, 3, or 12 | | Not assigned | any | Craps, user loses |
| Adds to 7 or 11 | | Not assigned | 1 | Natural, user wins |
| Adds to 4, 5, etc. (not craps or natural) | | Assigned to sum | 1 | Point set to sum, turn ends |
| Adds to 4, 5, etc. (not craps) | | Same as roll | 2+ | Same as point, user wins |

Each situation applies to the “computer players” as well, although user used as an example for test – very hard to define too many circumstances as the game seems relatively easy and basic (at least the simplified version)

**Sample Outputs:**

Graphical user interface, text, application, chat or text message

Description automatically generated

\*user is removed from the game after rolling craps/natural

Graphical user interface

Description automatically generated with medium confidence

\*if a player rolls craps on any turn they’re removed

Text, letter

Description automatically generated

\*naturals remove players, craps and natural can happen on the same roll

Text

Description automatically generated

\*rolling the same number that is your point gives you the win Text

Description automatically generated

\* first player to roll natural/their point is first place, followed by the proper order

Text

Description automatically generated