Table of Contents

Bluffer's Dice Game

Overview

Bluffer's Dice is a dice rolling game where you out-think or out-roll your opponent. It's all about the **highest roll**.

Players hide their rolls from each other during the game and this facilitates the guessing part of the game. Bluffing is allowed.

Because players hide their dice rolls from each other, the player must try to tell if their opponent is lying or telling the truth about their claim.

The object of the game is to call your opponent's bluff or to convince them to challenge your honest calls because proving your opponent wrong helps remove him from the game by giving you the point.

The game forces players to claim continuously higher rolls until someone finally issues a challenge to the other player.

Players

2 players

Equipment

10 dice, five for each player.
20 counters, ten for each player.

Game Terminology

These terms are used in descriptions of the game.

Dice – Standard six-sided dice.

Counter – Counting chips in the game to track each player's progress. Each player starts with a specific number of counters and these are depleted as the game progresses.

Pot – The holding container for the counters. When a player gives up a counter it goes back into the pot.

Caller – the current player.

Opponent – the other player.

Stand – A choice made by the player to keep the current roll.

Draw – A choice made by the player to roll again.

Claim – To state something whether it is true or not.

To Bluff – To pretend to have a good roll when you really don't.

Challenge – to accuse your opponent of bluffing (also know as "calling their bluff").

Hand – the kind of roll achieved across five dice. See the Hand Ranks table below.

Round – One cycle of the game which lasts until a player is challenged at which someone loses a counter.

Game Rules

Decide who Starts

- Start each player with 10 counters; all players must start with the same number. When a player loses all his counters he loses the game.
- To determine which player starts the game, each player rolls one die. The player that rolls the higher number starts the game. If there is a tie, the players roll again until someone achieves a higher number at which point the starting player is determined.

Game Play

Each round of play.

1. The caller rolls 5 dice but keeps the results hidden from their opponent. Usually this is accomplished by using a cup over the dice to shake them and obscure them from view.
2. The caller decides whether to "stand" (keep the current roll) or "draw" (roll again). If the caller chooses to draw then he may re-roll any of the five dice. The caller may choose to re-roll up to twice during the hand before announcing it.
3. The caller must announce a hand that is higher than the opponent's recent actual hand, whether that statement is true or not. The caller must be specific about her hand though. For example, the caller can't say in general, "*I have three of the same kind*." Rather the caller must be specific and state exactly what dice they have, for example: "*I have three fours*."
    i. See the "Hand Ranks" table below to see the ranking of hands relative to each other.

4. Move to the next player who becomes the caller and may make one of the following choices:
    i. Challenge the opponent's statement about her hand. At this point her dice are revealed and a winner is chosen based on who is correct. The loser of the round loses one counter to the pot.
    ii. Choose not to challenge the opponent and start his own roll. At this point go back to Step #1 and continue.

    Note: This interaction continues between the two players (the claims get higher) until someone issues a challenge to the other player and the round winner is decided.

5. Reveal your dice when challenged. If the challenger is wrong and you were honest about your hand (meaning you have the hand you claimed) then he loses the round and must put a counter into the pot. If you are caught bluffing (you don't have the hand you claimed) then you lose the round and must put one of your counters into the pot.

6. Begin a new round after a challenge is made.
7. Play many rounds. Win the game when you are the only one remaining who has any counters.

Hand Ranks (game roll combinations)

The following chart shows all possible game roll combinations using five dice. Rank #1 is the highest possible combination.

How to read this table:
Each letter represents any digit rolled on a dice. For example the Full House A-A-A-B-B represents two different numbers in the roll which could be 4-4-4-2-2 or 3-3-3-6-6 or 4-4-4-3-3 or 1-1-1-6-6, etc.

Rank	Roll name	Combination
1	Five of a kind	A-A-A-A-A
2	Four of a kind	A-A-A-A-B
3	Full house	A-A-A-B-B
	(is Three of a kind with a pair).	
4	Straight	A-B-C-D-E
	(Numbers must be in sequence.)	
5	Three of a kind	A-A-A-B-C
6	Two pairs	A-A-B-B-C
7	One pair	A-A-B-C-D
8	Runt	A-Z-S-U-F
	(All numbers are different.)	

To compare a claim against the most recent claim in the game (because each claim must be higher than the previous) compare them using the ranks in the table above to determine which is higher. **If two claims have the same rank,** for example, both claims are a Full house you can determine which claim is higher by the digits. Look at the triplets then pairs and so on. For example, a full house of **5-5-5**-3-3 beats a full house of **2-2-2**-6-6. A full house of 5-5-5-**3-3** beats a full house of 5-5-5-**2-2**.

Assignment Details

C#/.NET Implementation Suggestions

These are suggestions to port the game over to a C#/.NET program.

- Use data structures like C# arrays and collections; .NET FCL: System.Random, etc.
- The game requires player's rolls to be hidden from the other player. This is tough when the game is running on one computer screen. You can assume only one player will look at the screen at a time. You will want to build your screen for the current player and not give away the opponent's information.
- To pass the turn to the other player you might create a transition screen whereby the following actions take place:
    - The current player enters a command to finish his turn.
    - Your program displays a transition screen that instructs the current player to step away from the computer and for the opponent to approach.
    - The opponent must press a key to acquire his view of the game.

Assignment Learning Outcomes

You will apply knowledge from this course to achieve the outcomes of the assignment. Your work will be based on the .NET Framework, written using C# either as a Console or Windows application.

Assignment Learning Resources

Resources you will use to successfully complete this assignment are:

- Concepts learned and discussed in the classroom (attendance is important).
- Tutorials, code samples and reference materials from reference text books and information posted to the Seneca Blackboard system for the course.
- Microsoft MSDN documentation is helpful to research objects and members.

Access to Computers and Software

Time cannot be given to complete the assignment during class time. Assignments must be completed outside of the classroom. Seneca@York has general computer labs available in which you can work; the labs contain all necessary software. Some labs are open late.

Assignment Logistics and Due Date

Start early. If you start late you might not leave yourself enough time pass through the initial learning curve of coding use .NET before the assignment is due.

Each student must produce a unique assignment submission. Collaboration on ideas is encouraged. Copying code from other students is not allowed. You are allowed to repurpose any of your own code for your assignment, for example: lab work, personal

code. You can also take ideas and code samples posted to the Blackboard system if any proves helpful.

Your completed project is **due Week 7 (Thu Feb 28 by end of day)** and is worth 10% of your final mark. Hand in the following with your assignment submission:

- **Complete source code.**
- **A compiled and working program.**
  (A zipped/compressed project folder will contain this.)
  - Documentation is important. Helpful code comments must exist inside your program.
  - A well-tested application. Make a complete list of any bugs that you know about in your application and describe how you would go about correcting them.

A penalty of 15% per day late will be applied to overdue assignments. After one week the assignment will be worth no marks.

Submit your assignments electronically – printouts/hard copies are not required. Use one of the following means: attach it to an email, submit a DVD/CD-ROM or USB key. If you want to submit your assignment by other means please discuss with the instructor.

Extra Features for Bonus Marks

If you have included any extra features in your program and if you want to have them considered for marking, please submit the list of features with your assignment. Extra features must be "substantial" to be considered for bonus marks.

Assignment Grading

The following weights will be applied to grade your assignment submission. This assignment is worth 10% of your final course grade.

1. **Core Functionality** **70%**
   Does the application work according to functional requirements?

2. **User Interface** **10%**
   Intuitive and easy to use? A good/smooth experience?

3. **Classes/objects/methods** **10%**
   Appropriate use of the .NET Framework. Is the program organized in a modular fashion?

4. **Implementation Decisions** **10%**
   Determines if the coding is smart, efficient, tight. Is it commented and easy to understand? Are identifiers named well?