Chuck-A-Luck

by Richard Born
Associate Professor Emeritus
Northern Illinois University
rborn2@niu.edu

Introduction

Chuck-A-Luck is a very popular carnival and casino game. It is played with three dice that are contained in an hourglass shaped cage known as the *birdcage*, as shown in Figure 1. The game is called *crown and anchor* in British pubs. This is because the faces of the dice are inscribed with a diamond, heart, spade, club, crown and anchor. Elsewhere, ordinary six-sided dice with dots on the faces are used. The dice are rolled by the dealer giving the birdcage a spin.

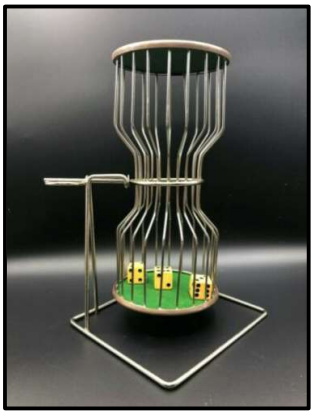

Figure 1

The rules for Chuck-A-Luck are really quite simple. The player selects an integer from 1 though 6. The three dice are then rolled. If exactly *n* dice show the player' number, then the payoff is *n* : 1. For example, suppose that a player bets 10 chips on the number 4. If none of the three dice show a 4, the payoff is 0. If one of the dice shows a 4, the payoff is 10. If two of the dice shows a 4, the payoff is 20. And is all three dice show a 4, the payoff is 30 chips.

This is a great game that can be used in a variety of disciplines in education, particularly *computer science* and *probability and statistics*. Computer science students can be challenged to design a program to simulate Chuck-A-Luck game play. Students of probability and statistics can use this game while studying combinations, random variables, and the binomial distribution. And, of course, anyone can learn about the risks involved when betting.

Chuck-A-Luck with Ozobot Evo

Figure 2 shows the Ozomap that is used with the accompanying OzoBlockly program *Chuck-A-Luck.ozocode*. Evo is your assistant in so many ways. He lets you place your bet. Through Evo, you specify your desired number from 1 through 6. He spins the birdcage for you, determining the results of the three dice. He is the dealer, making the payoff whenever you win. He is your statistician, keeping track of your winnings (or losses) as game play proceeds. Finally, he let's you stop playing when you've had enough in losses or when your pockets are full of winnings!

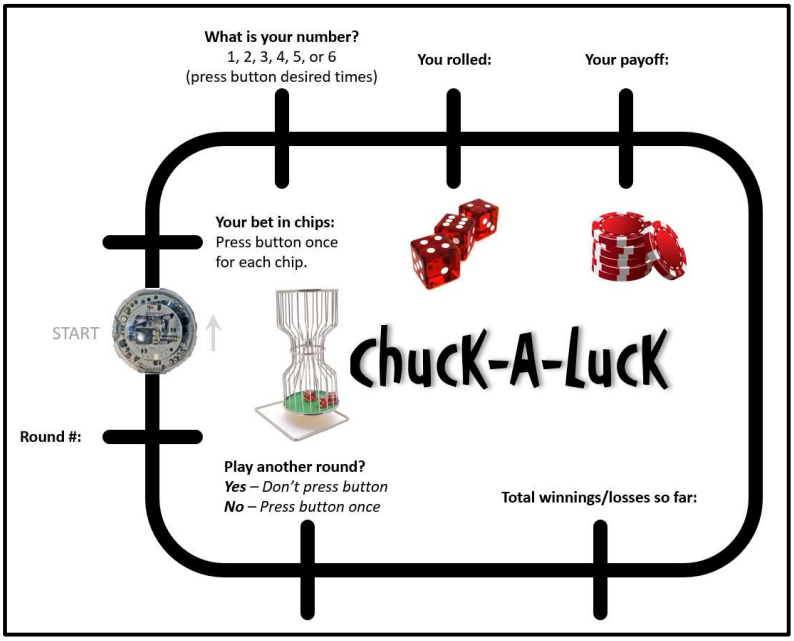

Figure 2

A full-page version of the Ozomap, suitable for using with Evo, can be obtained by printing the file *Chuck-A-Luck Ozomap.pdf*. Load the OzoBlockly program into Evo. Place Evo at the start location facing the direction shown by the arrow, and start the loaded program. When Evo's top light turns green, he moves to the betting location. You have 10 seconds to place your bet. Each press of Evo's button represents a bet of one chip. After each button press, Evo's top light will flash white quickly to confirm that the press was recognized. Evo then moves to the location where you specify your number from 1 to 6. Evo then moves to the location when the dice cage is spun, and Evo speaks the results of the three dice in the cage. Evo then moves to the payoff location, where he speaks the number of chips you get as a payoff. Next, Evo moves to the location where he indicates your total winnings (or losses, if negative) so far. Finally, Evo gives you a chance to play another round or quit.

Ozobot Chuck-A-Luck Just for Fun

Your students can play Ozobot Chuck-A-Luck just for fun—no need to know OzoBlockly programming. This will give them a qualitative feeling for the risks associated with betting and gambling without using money. They will find that most of the time, their payback is simply a recovery of the bet that they made. They will find that after, for example, 10 rounds of play, they are likely to be in the loss's column. They will probably discover that the odds of getting no payback at all on a given round are fairly high. They will discover that the odds of getting two of the dice to match their number are rather low. And they will probably wonder if their number will ever match all three dice!

Ozobot Chuck-A-Luck in a Statistics and Probability Class

Statistics and probability students commonly study what are known as *binomial distributions*. Suppose that an experiment consists of n independent binomial trials, each with a probability p of success and probability $q = 1 - p$ of failure. Then the probability that the experiment results in precisely x successes is given by the equation:

$$b(x; n, p) = \binom{n}{x} p^x q^{(n-x)}, \quad x = 0, 1, 2, 3, \dots, n.$$

In the Chuck-A-Luck game, $n = 3$, $p = \frac{1}{6}$, $q = (1 - \frac{1}{6}) = \frac{5}{6}$, and $x = \{0, 1, 2, 3\}$.

Note that $\binom{n}{x}$ represents the combinations of n things taken x at a time. We can quickly compute the probabilities, and will find the following:

$b(\mathbf{0}; 3, \frac{1}{6}) = \frac{125}{216}$ $b(\mathbf{1}; 3, \frac{1}{6}) = \frac{75}{216}$ $b(\mathbf{2}; 3, \frac{1}{6}) = \frac{15}{216}$ $b(\mathbf{3}; 3, \frac{1}{6}) = \frac{1}{216}$

Observations:

1. The probability of losing your bet on any given round of play ($x = \mathbf{0}$) is approximately 58%. Not very encouraging.
2. The probability of breaking even ($x = \mathbf{1}$) is about 35%. This is likely one thing that keeps you playing, since you get a payoff that covers the bet that you made.
3. The probability of your number matching two of the dice ($x = \mathbf{2}$) is about 7%. To put this into perspective, two matches occurs once in every 14 rounds of play, in the long run.
4. The probability of your number matching all three dice is about 0.5%. You would have to roll the dice about 200 times on average, before you would match all three!

Ozobot Chuck-A-Luck in a Computer Science Class

As an OzoBlockly programming project, design a program that will simulate 100 rounds of Chuck-A-Luck play. The only purpose of this program is for Evo to keep a count of the number of rounds for which there are 0 matches, 1 match, 2 matches, and 3 matches. No other statistics need be computed. Evo should remain motionless the entire time and speak the counts after the 100 rounds are completed.

Run the program 10 times recording the counts for each run in the table of Figure 3.

Run #	0 Match Count	1 Match Count	2 Match Count	3 Match Count
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
Totals→				
Experiment %				
Theoretical %				

Figure 3

Compare your experimental results with the theoretical results predicted in the previous section of this document.