If Josephus Had Played Russian Roulette

Hunter Glanz
Dr. Matt Carlton

Statistics Department
California Polytechnic State University
San Luis Obispo

June 12, 2009

Contents

1 Introduction

The *Josephus Problem* is a classic scenario used in many computer science and mathematics classes to help teach iteration, recursion, and modular arithmetic. The original problem dates back to ancient Rome. Josephus was a first century historian who was able to record the destruction of Jerusalem in AD 70. He actually fought the Romans in The First Jewish-Roman War (66-73 AD) as a Jewish military leader. After attacking and eventually cornering Josephus and his troops, the Romans asked him to surrender. Josephus' soldiers preferred suicide to capture and so Josephus devised a strategy for him and his comrades to die together. He suggested forming a circle and then proceeding to kill every third person. Josephus utilized some keen mathematical ability to discern where he and a friend should stand in order to be the last two standing. After doing so, the two of them surrendered to the Romans and later Josephus became known as Titus Flavius Josephus as a new Roman citizen.

The more general problem, or game, involves n people in a circle with a particular skip number, k. Starting at position 1, we move forward k places and eliminate the player at that position. At each step in the game, only the "living" players are considered when skipping ahead. The game ends when there is only one person left "alive".

Example

7 players	$n = 7$
Skip number is 3	$k = 3$

$$\begin{array}{ccccccc}
\underline{1} & \underline{2} & \boxed{3} & \underline{4} & \underline{5} & \underline{6} & \underline{7} \\
\underline{4} & \underline{5} & \boxed{6} & \underline{7} & \underline{1} & \underline{2} & \\
\underline{7} & \underline{1} & \boxed{2} & \underline{4} & \underline{5} & & \\
\underline{4} & \underline{5} & \boxed{7} & \underline{1} & & & \\
\underline{1} & \underline{4} & \boxed{5} & & & & \\
\boxed{1} & \underline{4} & & & & & \\
\underline{4} & & & & & &
\end{array}$$

Here we have strung out the circle of people into a line. The boxed player in each row (or each step) is the one being eliminated. At each step of the game, players eliminated during any previous turns have been removed. Also, the "living" players have been reordered so that the counting always starts on the far left: 1, 2, 3 (since 3 is the skip number). During the second to last turn, when only the first and fourth players remain, we must cycle back through them while counting. Hence, to determine that the first player was eliminated and the fourth player was the winner we counted 1, 4, 1.

Mapping out a game like the one above seems easy enough, but for larger n and k the winner could become more difficult to determine. The most important question is: does there exist a closed formula for the winning position in a game of n people with skip number k? As I stated before, the problem arises in a number of subject areas and with the abundance of attention a few results have been reached. Let $J(n, k)$ denote the winning position in the Josephus Game with n players and skip number k. When the skip number is 2, there is indeed a formula for the winning position. Write $n = 2^a + t$, where 2^a is the highest power of 2 such that $2^a < n$. Then,

$$J(2^a + t, 2) = 2t + 1$$

This solution as well as a proof can be found at [**?**] under Games.

Example 1

7 players	$n = 7$
Skip number is 2	$k = 2$

$\Rightarrow 7 = 2^2 + 3$
$\Rightarrow a = 2, t = 3$

So, we have

$$J(2^2 + 3, 2) = 2(3) + 1 = 7$$

Thus the winner is the seventh player, $J(7, 2) = 7$.

Example 2

49 players	$n = 49$
Skip number is 2	$k = 2$

$\Rightarrow 49 = 2^5 + 17$
$\Rightarrow a = 5, t = 17$

So, we have

$$J(2^5 + 17, 2) = 2(17) + 1 = 35$$

Thus the winner is the thirty-fifth player, $J(49, 2) = 35$.

In the general case with n people and skip number k, the solution is a recurrence of the form:

$$J(n+1,k) \equiv (J(n,k)+k) \bmod (n+1), \quad \text{with } J(1,k)=0$$

This recursive solution can be observed by considering a particular n and k for which the winner, $J(n,k)$, is known. Then consider the same game with n+1 people and eliminate the first person. We have now arrived back at the original game for which the winner was known. By rotating the remaining positions, it is simple to pick out the known winner and hence $J(n+1,k)$. See the example below.

First Game: $n = 5, \quad k = 2$

Original Order	1	2	3	4	5
Order Eliminated	2	4	1	5	**3**

Hence the winner is the third player: $J(5,2)=3$.

Second Game: $n = 6, \quad k = 2$

Original Order	1	2	3	4	5	6
Order Eliminated	2	4	6	3	1	**5**
Reordered After First Elimination	3	4	**5**	6	1	

After the first elimination, the five remaining players (all those but 2) were reordered to display their line-up in a five person game (such as our first game). But we know the winner to this scenario: the third player. As demonstrated by the bold faced players, the winner could have been arrived at either of these two ways. Thus $J(6,2)$ is the fifth player.

$$\begin{aligned} J(6,2) &\equiv (J(5,2)+2) \bmod 6 \\ &\equiv (3+2) \bmod 6 \\ &= 5 \end{aligned}$$

While this algorithm works, it is a recursive solution and so far no analysis has produced a closed formula for the winner, $J(n,k)$, of a general game. For the remainder of this paper, the total number of people in a particular Josephus Game will be denoted n, and the skip number will be denoted k.

2 Variations

While the original Josephus Game is quite interesting and a very challenging mathematical scenario, many other questions pop up when pondering the classic problem in a contemporary context. For example, what if instead of everyone in the circle taking a sword to themselves in turn, there were a designated shooter, who shot every k^{th} person? Furthermore, what if that shooter had limited accuracy and only had probability p of actually hitting a player on any given turn? Finally, what if it took more than one shot to kill (eliminate) a player?

We have just defined two new parameters for the Josephus Game: the probability, p, of being eliminated and the number of shots, s, it takes to eliminate a player. These questions will be addressed in the remainder of this article.

2.1 Probabilistic Josephus Game

In this section, we will discuss the implementation of the parameter p into the original game. So, for the time being, it will only take one shot to eliminate a player. It is important to observe that when $k = 1$ we are in some sense playing Russian Roulette. For this section, let $P(w; n, k, p)$ denote the probability of the w^{th} player ($w = 1, \ldots, n$) winning in the Josephus Game with n players, skip number k, and probability of being eliminated p. Though this may seem confusing at first, we must change our notation as well as our perspective of the game. There no longer exists a fixed winner for a game with particular n and k. Because there is now a probability associated with being eliminated at each step of the game, there is now a probability of winning the game for each player. Hence our new notation, $P(w; n, k, p)$.

In Russian Roulette, there are n people in a circle who take turns sequentially, firing a gun at themselves. In terms of the variables we have defined, this means that in Russian Roulette $k = 1$. The gun is not fully loaded, so there is in fact a probability less than 1 of being eliminated (killed). The game ends when there is only one person left standing, just as in the original Josephus game. Quite a bit of work has been done mathematically analyzing Russian Roulette, and the results we'll emphasize are found in Blom et. al. [**?**]. They were able to produce an explicit solution to this particular scenario:

$$P(w; n, 1, p) = p\sum_{j=1}^{\infty}(1 - q^{j+1})^{w-1}(1 - q^j)^{n-w}q^j$$

for $w = 1, \ldots, n - 1$. In the case that $w = n$, the expression changes slightly to the following:

$$P(n; n, 1, p) = p\sum_{j=0}^{\infty}(1 - q^{j+1})^{w-1}(1 - q^j)^{n-w}q^j$$

Once dissected, the parts of this formula can be derived using the Geometric Distribution with parameter p. Let B_j be the event that player w gets shot in the $(j+1)$st turn, where $j = 0, 1, \ldots$. The events B_j are disjoint and $P(B_j) = q^j p$. If B_0 occurs, then player w can never win. Given that B_j, $j > 0$, occurs, player w wins if the following events C_j and D_j occur:

C_j: Players 1, 2, ..., $w - 1$ get shot before player w (in the $(j+1)$st step or earlier).
D_j: Players $w + 1$, $w + 2$, ..., n get shot before player w (in the j^{th} step or earlier).

It is these three independent events that have a geometric distribution. They establish that the probability that player 1 gets shot on or before the $(j+1)$st step is $1 - q^{j+1}$. Consequently,

$$P(C_j) = (1 - q^{j+1})^{w-1}$$

$$P(D_j) = (1 - q^j)^{n-w}$$

Thus

$$P(w; n, 1, p) = \sum_{j=1}^{\infty} P(B_j C_j D_j) = \sum_{j=1}^{\infty} P(B_j)P(C_j)P(D_j)$$

From here we arrive at the initial formulation above. To help demonstrate this result, consider the following example:

<u>$\boldsymbol{n = 3,\ k = 1,\ p = \frac{2}{3}}$</u>

$$\begin{aligned}
P(1; 3, 1, \tfrac{2}{3}) &= \tfrac{2}{3}\textstyle\sum_{j=1}^{\infty}(1 - \tfrac{1}{3}^{j+1})^{1-1}(1 - \tfrac{1}{3}^{j})^{3-1} \cdot \tfrac{1}{3}^{j} \\
&= \tfrac{2}{3}\textstyle\sum_{j=1}^{\infty}(1 - \tfrac{1}{3}^{j})^{2} \cdot \tfrac{1}{3}^{j} \\
&= \tfrac{2}{3}\textstyle\sum_{j=1}^{\infty}(1 - 2(\tfrac{1}{3})^{j} + (\tfrac{1}{3})^{2j}) \cdot \tfrac{1}{3}^{j} \\
&= \tfrac{2}{3}\textstyle\sum_{j=1}^{\infty}[(\tfrac{1}{3})^{j} - 2(\tfrac{1}{3})^{2j} + (\tfrac{1}{3})^{3j}] \\
&= \tfrac{2}{3}\left[\textstyle\sum_{j=1}^{\infty}(\tfrac{1}{3})^{j} - 2\sum_{j=1}^{\infty}(\tfrac{1}{9})^{j} + \sum_{j=1}^{\infty}(\tfrac{1}{27})^{j}\right] \\
&= \tfrac{2}{3}\left[\tfrac{1}{2} - \tfrac{1}{4} + \tfrac{1}{26}\right] \\
&= \tfrac{5}{26}
\end{aligned}$$

$$\begin{aligned}
P(2; 3, 1, \tfrac{2}{3}) &= \tfrac{2}{3}\textstyle\sum_{j=1}^{\infty}(1 - \tfrac{1}{3}^{j+1})^{2-1}(1 - \tfrac{1}{3}^{j})^{3-2} \cdot \tfrac{1}{3}^{j} \\
&= \tfrac{2}{3}\textstyle\sum_{j=1}^{\infty}(1 - \tfrac{1}{3}^{j+1})(1 - \tfrac{1}{3}^{j}) \cdot \tfrac{1}{3}^{j} \\
&= \tfrac{2}{3}\textstyle\sum_{j=1}^{\infty}(1 - (\tfrac{1}{3})^{j+1} - (\tfrac{1}{3})^{j} + (\tfrac{1}{3})^{2j+1}) \cdot \tfrac{1}{3}^{j} \\
&= \tfrac{2}{3}\left[\textstyle\sum_{j=1}^{\infty}(\tfrac{1}{3})^{j} - \tfrac{1}{3}\sum_{j=1}^{\infty}(\tfrac{1}{9})^{j} - \sum_{j=1}^{\infty}(\tfrac{1}{9})^{j} + \tfrac{1}{3}\sum_{j=1}^{\infty}(\tfrac{1}{27})^{j}\right] \\
&= \tfrac{2}{3}\left[\tfrac{1}{2} - \tfrac{1}{24} - \tfrac{1}{8} + \tfrac{1}{78}\right] \\
&= \tfrac{3}{13}
\end{aligned}$$

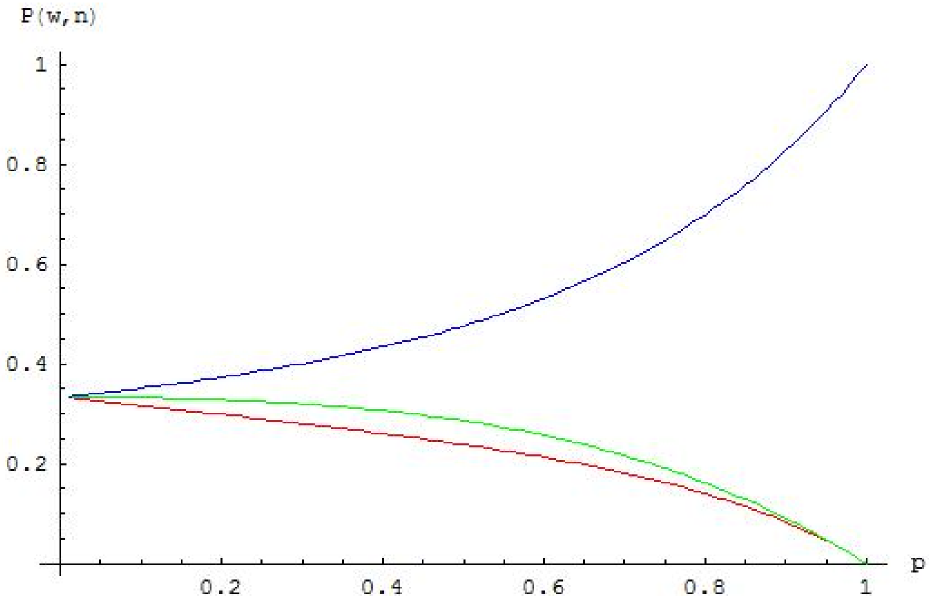
$$\begin{aligned}
P(3; 3, 1, \tfrac{2}{3}) &= \tfrac{2}{3}\textstyle\sum_{j=0}^{\infty}(1 - \tfrac{1}{3}^{j+1})^{3-1}(1 - \tfrac{1}{3}^{j})^{3-3} \cdot \tfrac{1}{3}^{j} \\
&= \tfrac{2}{3}\textstyle\sum_{j=0}^{\infty}(1 - \tfrac{1}{3}^{j+1})^{2} \cdot \tfrac{1}{3}^{j} \\
&= \tfrac{2}{3}\textstyle\sum_{j=0}^{\infty}(1 - 2(\tfrac{1}{3})^{j+1} + (\tfrac{1}{3})^{2j+2}) \cdot \tfrac{1}{3}^{j} \\
&= \tfrac{2}{3}\left[\textstyle\sum_{j=0}^{\infty}(\tfrac{1}{3})^{j} - \tfrac{2}{3}\sum_{j=0}^{\infty}(\tfrac{1}{9})^{j} + \tfrac{1}{9}\sum_{j=0}^{\infty}(\tfrac{1}{27})^{j}\right] \\
&= \tfrac{2}{3}\left[\tfrac{3}{2} - \tfrac{3}{4} + \tfrac{3}{26}\right] \\
&= \tfrac{15}{26}
\end{aligned}$$

So, in this scenario the third player is the most likely to win. In general if we leave the parameter p in the above three expressions we can find the probability of each of the three players winning as a function of p. You can see from the functions and graph below who is most likely to win for various values of p (Player 1 is red, Player 2 is green, and Player 3 is blue).

$$P(1;3,1,p) = \frac{-2+4p-3p^2+p^3}{-6+9p-5p^2+p^3}$$

$$P(2;3,1,p) = \frac{1-p}{3-3p+p^2}$$

$$P(3;3,1,p) = \frac{2-2p+p^2}{6-9p+5p^2-p^3}$$



In light of Josephus' scenario classic Russian Roulette is only the root of the probabilistic version of the Josephus Game. Without a closed formula for the original game, I initially worked towards creating a Josephus Game program in Mathematica that could simulate a game with all of the parameters I've discussed so far (all Mathematica code can be found in the Appendix). Because of the simplicity of the program and the high power of the computers available, I could simulate hundreds of thousands of games in order to determine the long term frequencies of winning positions. Even with these new capabilities, I was a bit unsatisfied. It was not until later that we (Dr. Carlton and myself) discovered a way to explicitly calculate the probability of winning for each player in a particular game. Consider the following:

Let p denote the probability of being eliminated on a given step ($q = 1 - p$) and let k denote the skip step. Suppressing these arguments momentarily, let $P(w, n)$ denote the probability that player number w is the winner in an n player Josephus Game. Here we'll be considering different values of the n and k parameters in completely self-contained scenarios. To make the following postulations imagine we're at the first step of the game about to take the first shot. That first shot can either be a success or failure, and the probability that each player wins the game will be affected by the outcome of this first shot. Follow along in the very first scenario below as I dissect this process.

***k* = 1**

n = 2

$$P(1,2) = p \cdot 0 + q \cdot P(2,2)$$

In this scenario the first player to be shot at is player one. If the shot is successful (with probability p), then player one has probability 0 of winning the game; hence the $p \cdot 0$. If the shot is unsuccessful (with probability q), then the second player will be shot at next. This makes player one the "second player" in this subsequent two-player game. Thus, his probability of winning has become $P(2,2)$. We have now accounted for the two possible outcomes of the first shot and can write $P(1,2) = q \cdot P(2,2)$.

$$P(2,2) = p \cdot 1 + q \cdot P(1,2)$$

Again, we know that the first player to be shot at is player one. If the shot is successful, then player two has probability 1 of winning the game since player one has been eliminated; hence the $p \cdot 1$. If the shot is unsuccessful, then the second player will be shot at next. This makes player two the "first player" in this subsequent two-player game. Thus, his probability of winning has become $P(1,2)$. Once again we have accounted for the two possible outcomes of the first shot and can write $P(2,2) = p \cdot 1 + q \cdot P(1,2)$. Notice that we now have two equations and two unknowns ($P(1,2)$ and $P(2,2)$).

So, we have

$$\begin{array}{rrl} P(1,2) - & q \cdot P(2,2) & = 0 \\ -q \cdot P(1,2) + & P(2,2) & = p \end{array}$$

which we can write

$$\begin{bmatrix} 1 & -q \\ -q & 1 \end{bmatrix} \begin{bmatrix} P(1,2) \\ P(2,2) \end{bmatrix} = \begin{bmatrix} 0 \\ p \end{bmatrix}$$

n = 3

By the same logic, we can write a set of relationships based upon the first shot in a three-player game.

$$\begin{array}{rrr} P(1,3) = & p \cdot 0 + & q \cdot P(3,3) \\ P(2,3) = & p \cdot P(1,2) + & q \cdot P(1,3) \\ P(3,3) = & p \cdot P(2,2) + & q \cdot P(2,3) \end{array}$$

So, we have

$$\begin{aligned} P(1,3) - \quad & q{\cdot}P(3,3) &= 0 \\ -q{\cdot}P(1,3) + \quad & P(2,3) &= p{\cdot}P(1,2) \\ -q{\cdot}P(2,3) + \quad & P(3,3) &= p{\cdot}P(2,2) \end{aligned}$$

which we can write

$$\begin{bmatrix} 1 & 0 & -q \\ -q & 1 & 0 \\ 0 & -q & 1 \end{bmatrix} \begin{bmatrix} P(1,3) \\ P(2,3) \\ P(3,3) \end{bmatrix} = p \cdot \begin{bmatrix} 0 \\ P(1,2) \\ P(2,2) \end{bmatrix}$$

Note that above I have outlined the algorithm in the Russian Roulette case ($k = 1$). In any case, it is a good place to start. As you can see, we can solve for $P(1,2)$ and $P(2,2)$ in the $n = 2$ case using matrix algebra. Using these computed values we can then solve for $P(1,3)$, $P(2,3)$, and $P(3,3)$ in the $n = 3$ case using the same method. With this recursion we can solve for any $P(w; n, 1, p)$. This algorithm provides an excellent foundation from which to attack the scenario where skip = k. Consider the following:

***k* = 2**

$n = 2$

$$\begin{aligned} P(1,2) &= 1 \\ P(2,2) &= 0 \end{aligned}$$

$n = 3$

$$\begin{aligned} P(1,3) &= p{\cdot}P(2,2) + q{\cdot}P(2,3) \\ P(2,3) &= p{\cdot}0 + q{\cdot}P(3,3) \\ P(3,3) &= p{\cdot}P(1,2) + q{\cdot}P(1,3) \end{aligned}$$

So, we have

$$\begin{aligned} P(1,3) - \quad & q{\cdot}P(2,3) &= p{\cdot}P(2,2) \\ -q{\cdot}P(3,3) + \quad & P(2,3) &= 0 \\ -q{\cdot}P(1,3) + \quad & P(3,3) &= p{\cdot}P(1,2) \end{aligned}$$

which we can write

$$\begin{bmatrix} 1 & -q & 0 \\ 0 & 1 & -q \\ -q & 0 & 1 \end{bmatrix} \begin{bmatrix} P(1,3) \\ P(2,3) \\ P(3,3) \end{bmatrix} = p \cdot \begin{bmatrix} P(2,2) \\ 0 \\ P(1,2) \end{bmatrix} \tag{1}$$

Observe that this last line can also be written

$$\begin{bmatrix} 1 & -q & 0 \\ 0 & 1 & -q \\ -q & 0 & 1 \end{bmatrix} \begin{bmatrix} P(1,3) \\ P(2,3) \\ P(3,3) \end{bmatrix} = p \cdot \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} P(1,2) \\ P(2,2) \\ 0 \end{bmatrix} \tag{2}$$

Again, note that using this same type of recursion we can determine any $P(w; n, 2, p)$. At this stage it is important to note that the matrices involved in both of the above cases ($k = 1, 2$) are circulant matrices. A circulant matrix is a special kind of Toeplitz matrix where each row vector is rotated one element to the right relative to the preceding row vector. In general, circulant matrices are of the following form:

$$C = \begin{bmatrix} c_0 & c_{n-1} & \dots & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & & c_2 \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ c_{n-2} & & \ddots & \ddots & c_{n-1} \\ c_{n-1} & c_{n-2} & \dots & c_1 & c_0 \end{bmatrix}$$

With the use of circulant matrices we can rewrite the solution vector in (**??**) as it's displayed in (**??**). In fact, we can rewrite the left-hand side of (**??**) using the same circulant matrix.

$$\left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - q \cdot \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \right) \begin{bmatrix} P(1,3) \\ P(2,3) \\ P(3,3) \end{bmatrix} = p \cdot \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} P(1,2) \\ P(2,2) \\ 0 \end{bmatrix} \tag{3}$$

The ability to rewrite our recursion in this way is neat, but is it predictable? Yes! Note that the circulant matrix used above in the $n = 3$ and $k = 2$ case can be obtained by rotating each row of the 3x3 identity matrix to the left 2 cells; or if you prefer, you could rotate each column down 2 cells. Indeed, if you look back at our short look at the $k = 1$ scenario on page 5 you'll discover that a similar circulant matrix could be used to rewrite that recursion. However, in the $k = 1$ case, each row is rotated only 1 cell to the left. More examples would easily confirm that these circulant matrices correspond to the skip number in each situation. In a Josephus Game of n people and skip number k, the appropriate circulant matrix could be obtained by rotating each row of the nxn identity matrix to the left k cells.

We now have the tools to determine any $P(w; n, k, p)$. We need only solve the following system using the recursion we've outlined above:

$$(I_n - q \cdot R_{n,k}) \begin{bmatrix} P(1,n) \\ P(2,n) \\ \vdots \\ P(n,n) \end{bmatrix} = p \cdot R_{n,k} \begin{bmatrix} P(1,n-1) \\ P(2,n-1) \\ \vdots \\ P(n-1,n-1) \\ 0 \end{bmatrix}$$

where I_n is the $n \times n$ identity matrix, $R_{n,k}$ is the $n \times n$ circulant matrix corresponding to skip number k, and q is $1-p$. With the power of modern computers we are equipped to rigorously verfiy these results through simulation. Using Mathematica and SAS, I implemented the algorithm derived above as well as a program useful for simulating the Josephus Game and storing the results. Below are results comparing what our recursion produced to long run frequencies observed through the simulation. In these simulations 1,000,000 games were played with the specified parameters and the winner recorded each time.

***n* = 6, *k* = 4, *p* = .25**

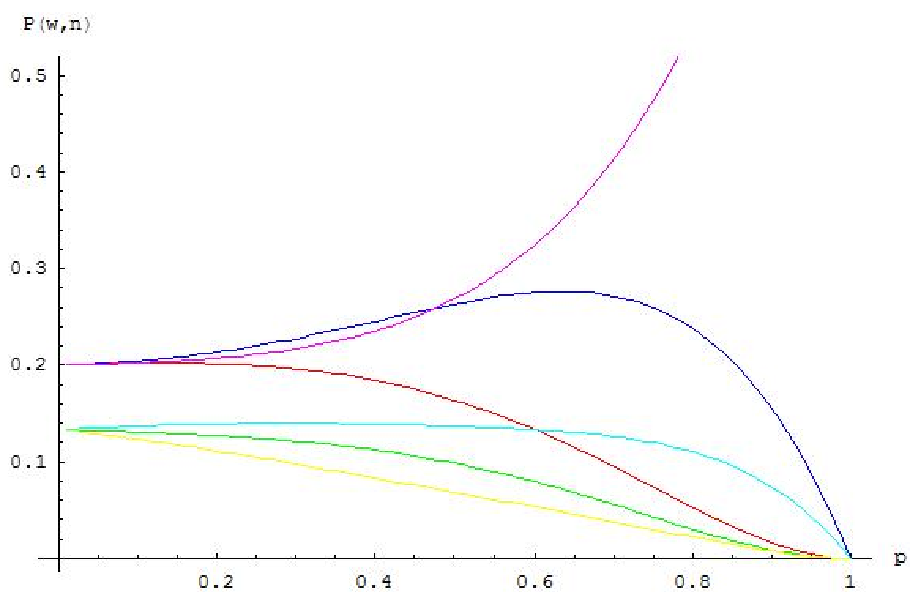
	$P(1;6,4,.25)$	$P(2;6,4,.25)$	$P(3;6,4,.25)$	$P(4;6,4,.25)$	$P(5;6,4,.25)$	$P(6;6,4,.25)$
Recursion	0.1995	0.1247	0.2200	0.1048	0.2113	0.1396
Simulation	0.1997	0.1241	0.2196	0.1048	0.2121	0.1396

***n* = 5, *k* = 7, *p* = .83**

	$P(1;5,7,.83)$	$P(2;5,7,.83)$	$P(3;5,7,.83)$	$P(4;5,7,.83)$	$P(5;5,7,.83)$
Recursion	0.2040	0.0097	0.2188	0.5104	0.0571
Simulation	0.2036	0.0096	0.2181	0.5116	0.0571

Though these were just two arbitrary scenarios, the results are extremely close. The probabilities produced by the recursion should be considered exact after observing such small deviances using simulation. Thus our algorithm can indeed determine any $P(w;n,k,p)$. Mathematica's inherently symbolic environment gives us a significant advantage in viewing $P(w;n,k,p)$ as a function of p. Consider the same two scenarios as above, but let p vary from 0 to 1.

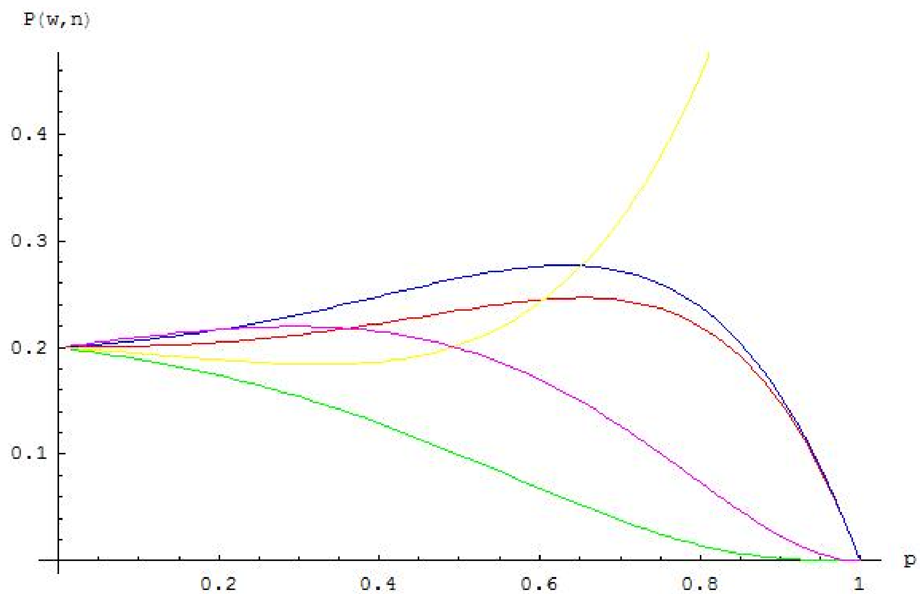
***n* = 6, *k* = 4**



For example, the probability of the fifth player winning the game as a function of p is

$$P(5;6,4,p) = \frac{9 - 35p + 66p^2 - 69p^3 + 42p^4 - 14p^5 + 2p^6}{(3 - 3p + p^2)^2(5 - 10p + 10p^2 - 5p^3 + p^4)}$$

<u>$\boldsymbol{n = 5, k = 7}$</u>



Similarly, the probability of the fourth player winning the game as a function of p is

$$P(4;5,7,p) = -\frac{4 - 15p + 26p^2 - 24p^3 + 14p^4 - 5p^5 + p^6}{(-2 + p)(2 - 2p + p^2)(5 - 10p + 10p^2 - 5p^3 + p^4)}$$

It is great to be able to visualize these results. We can also use the graphs to verify things we already had an idea about. For example, we know that the winners of the deterministic ($p = 1$) Josephus Games above are the fifth and fourth players, respectively. So it makes sense that the curves representing their probabilities of winning approach 1 as p goes to 1. For the same reason, it makes sense that all of the other curves go to 0 as p goes to 1. Since 6 and 4 are not relatively prime the second, fourth and sixth players in first scenario ($n = 6, k = 4$) will continue to be shot at without interruption until one of them is eliminated. This is why half of the players for this scenario start out at a different probability than the other half for p close to 0. On the other hand, since 5 and 7 are relatively prime, in the second scenario ($n = 5, k = 7$) no one is destined to be eliminated before the rest. Hence, all five players start out with the same probability of winning for p close to 0.

To conclude this section on the probabilistic version of the Josephus Game observe that we can use the recursion developed above with the stipulation that $p = 1$ to solve the deterministic Josephus Game. When the probability of being eliminated is 1 we've arrived back at the classic Josephus scenario. Mathematically,

$$\lim_{p \to 1^-} P(w; n, k, p) = \begin{cases} 0 & \text{if } J(n,k) = w \\ 1 & \text{if } J(n,k) \neq w \end{cases}$$

2.2 Multiple Wounds

In this section, we will discuss the implementation of the s parameter into the original Josephus Game. Up until we have been discussing the case where $s = 1$ (i.e. it takes only one shot to eliminate a player). Intuitively it seems that if $s \neq 1$, then the game and thus the winner would be different. However, n and k still heavily influence everything. For this section, let $J(n, k, s)$ denote the winning position of a Josephus Game with n people, skip number k, and s shots to eliminate a player. Again, we are changing the language to reflect the effect of the current parameters on the original game. In this section we address the deterministic game ($p = 1$) and proceed to analyze the original Josephus Game where it takes s shots to eliminate a player. Thus, there will be a unique winner for every combination of parameter values.

It turns out that if n and k are relatively prime (i.e. $gcd(n, k) = 1$), then it does not matter what s is. The order the players are eliminated and the winner of the game remain the same in this case. I've illustrated this in the following two games.

$n = 8$, $k = 4$

Original Order	1	2	3	4	5	6	7	8
Order Eliminated	4	8	5	2	1	3	7	6

In this scenario the winner is the sixth player, $J(8, 4, 1) = 6$. Here n and k are not relatively prime, which contextually means that some players are destined to be eliminated before the rest. Observe how the game changes when we change s from 1 to 2.

Original Order	1	2	3	4	5	6	7	8								
Order Shot	4	8	4	8	5	2	7	5	2	1	1	3	6	7	6	3
Order Eliminated	4	8	5	2	1	7	6	3								

When $s = 2$, the winner is the third player; $J(8, 4, 2) = 3$. The fourth and eighth players are doomed to die before the rest; and notably before the other players have even been hit once. In fact, the winner changes again if we increase s to 3. This is all linked to the fact that $gcd(8, 4) = 4$. On the other hand, consider the next example where n and k are relatively prime.

$n = 7$, $k = 3$

Original Order	1	2	3	4	5	6	7
Order Eliminated	3	6	2	7	5	1	4

In this scenario the winner is the fourth player, $J(7, 3, 1) = 4$. As opposed to the first scenario, in this one no player is doomed to be eliminated before the rest. Consequently, the value of s has no effect on the outcome of this game. Observe the special thing that happens when s is increased in the case that n and k are relatively prime.

Original Order	1	2	3	4	5	6	7							
Order Shot	3	6	2	5	1	4	7	3	6	2	7	5	1	4
Order Eliminated	3	6	2	7	5	1	4							

Because n and k are relatively prime, the order in which the players are shot during the final round of shots is the same as in the $s = 1$ case. In other words, every player receives their eliminating shot in the same order as they would in the original $s = 1$ game. This does mean that in the $s = 1$ case, the k^{th} player is indeed doomed to be eliminated first. The players do not receive *every* shot in the order that they get eliminated. However, they do receive the shots in a way such that no player is repeatedly shot at more than any other. We can observe this phenomenon without going through Josephus Games using brute force.

$n = 8$, $k = 4$

$$\begin{aligned} & 4 \cdot \{1, 2, 3, 4, 5, 6, 7, 8\} \\ = \; & \{4, 8, 12, 16, 20, 24, 28, 32\} \end{aligned}$$

Now, if we mod this list (the set of our positions multiplied by $k = 4$) by $n = 8$ we'll arrive at a reordering of our positions (treating a 0 as n). This will precisely be the order in which they will be shot!

$$\begin{aligned} & \{4, 8, 12, 16, 20, 24, 28, 32\} \bmod 8 \\ = \; & \{4, 8, 4, 8, 4, 8, 4, 8\} \end{aligned}$$

Again, what we have done is take a list of our players, multiplied through by $k = 4$ and mod by $n = 8$. We know that we'll be aiming at every fourth player until all eight players have been eliminated. In a sense we've shuffled our list of players according to this knowledge. Our resulting list is telling us that none of the other players will be eliminated until either players 4 or 8 have been eliminated. At least one of these players is doomed to be eliminated before the rest are even shot once. Compare this to our second scenario from above.

$n = 7$, $k = 3$

$$\begin{aligned} & 3 \cdot \{1,2,3,4,5,6,7\} \\ = & \{3,6,9,12,15,18,21\} \\ \\ & \{3,6,9,12,15,18,21\} \bmod 7 \\ = & \{3,6,2,5,1,4,7\} \end{aligned}$$

Notice here that even though this is not the order that the players are eliminated, it is the order that the players will be shot until the first player is eliminated. Because this new list is just a reordering of all the original players we can see that there are no players doomed to die before the rest. Each player will receive their first shot before anyone else receives their second; and so forth.

To conclude this section, note that we were unable to formulate an algorithm to determine any particular $J(n,k,s)$. The best that we could do is actually run through a game with particular n, k, and s in order to determine the winner. We are not empty-handed though. We established that if n and k are relatively prime then the value of s does not matter, and the winner is actually the same as in the game with n people, skip number k, and $s = 1$. Our lingering question is whether a recursion exists for the $s>1$ case.

2.3 Multiple Probabilistic Wounds

In this section we'll include both the p and s parameters to form a third variant of the Josephus Game. Again, because there is a probability of being eliminated there will be no clear winner of a particular game. Instead we'll refer to $P(w;n,k,s,p)$ as the probability of the w^{th} player winning the Josephus Game with n people, skip number k, probability of being eliminated p, and s shots to be eliminated.

With the formulation of this new problem, it makes sense to start back at the most basic forms this game could take on. We've considered the case where $s = 1$. We've considered the case where $p = 1$. Now we'll let both of these parameters vary along with n and fix $k = 1$. In other words, we will start by looking at Russian Roulette when it takes s shots to eliminate a player.

Observe the following analogy. The expression for $P(w;n,1,1,p)$ formulated by Blom et. al. used a combination of events, each of which followed a geometric distribution. When we allow s to take on values other than 1, these events should be thought in the context of being shot the s^{th} time and not just the once. Consequently, these new events will follow a negative binomial distribution instead of a geometric distribution. We generalized the original expression to this new scenario using the probability mass function of the negative binomial distribution.

$$P(w;n,1,s,p)=\sum_{j=1}^{\infty}\left(\sum_{x=s}^{j+1}\binom{x-1}{s-1}p^sq^{x-s}\right)^{w-1}\left(\sum_{x=s}^{j}\binom{x-1}{s-1}p^sq^{x-s}\right)^{n-w}\left(\binom{j}{s-1}p^sq^{j+1-s}\right)$$

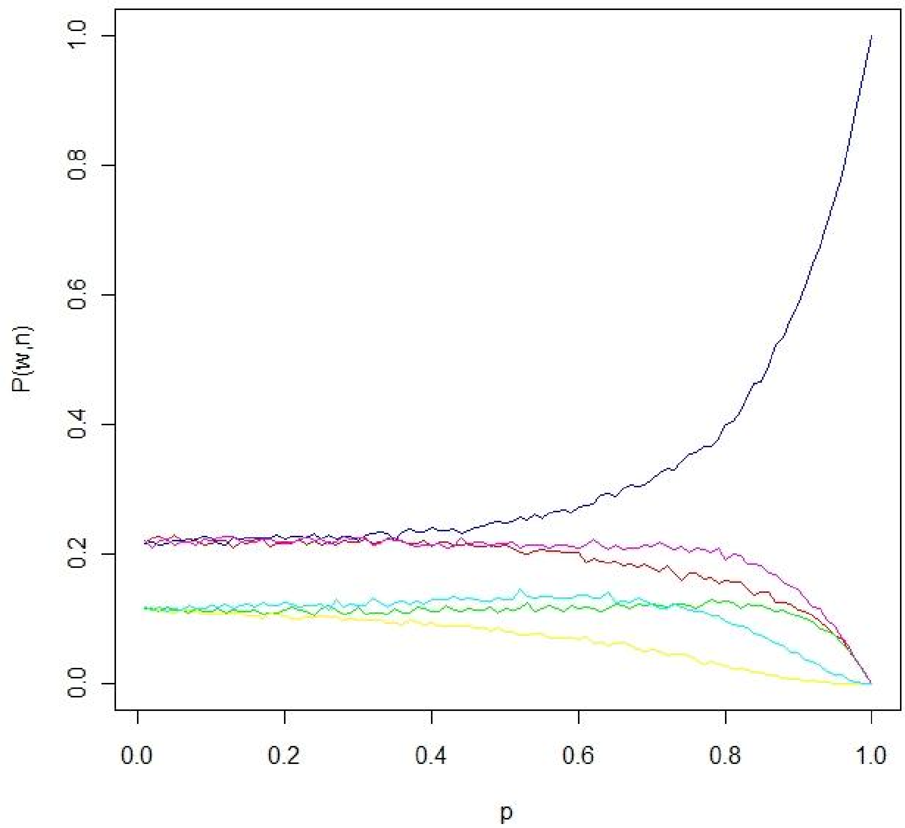
Note again that this expression only holds for $w=1,\ldots,n-1$. The infinite sum must start at $j=0$ when $w=n$ as in the original case. This expression is quite ugly. However, with the help of mathematical software we can verify this result. If we set $s=1$ and try any other values for n, k, and p we can use both the original Russian Roulette expression and our matrix recursion from above to check this result. Without pouring through the details involved in determining even one player's probability of winning with this new expression, this equation will simplify back to the original and produce the correct answers. But how do we know it works for $s\neq 1$? We can simulate the results and look at long run frequencies!

$n=6$, $k=1$, $s=2$, $p=\frac{1}{2}$

	$P(1;6,1,2,.5)$	$P(2;6,1,2,.5)$	$P(3;6,1,2,.5)$	$P(4;6,1,2,.5)$	$P(5;6,1,2,.5)$	$P(6;6,1,2,.5)$
Expression	0.1304	0.1413	0.1542	0.1699	0.1894	0.2147
Simulation	0.1295	0.1409	0.1534	0.1721	0.1891	0.2150

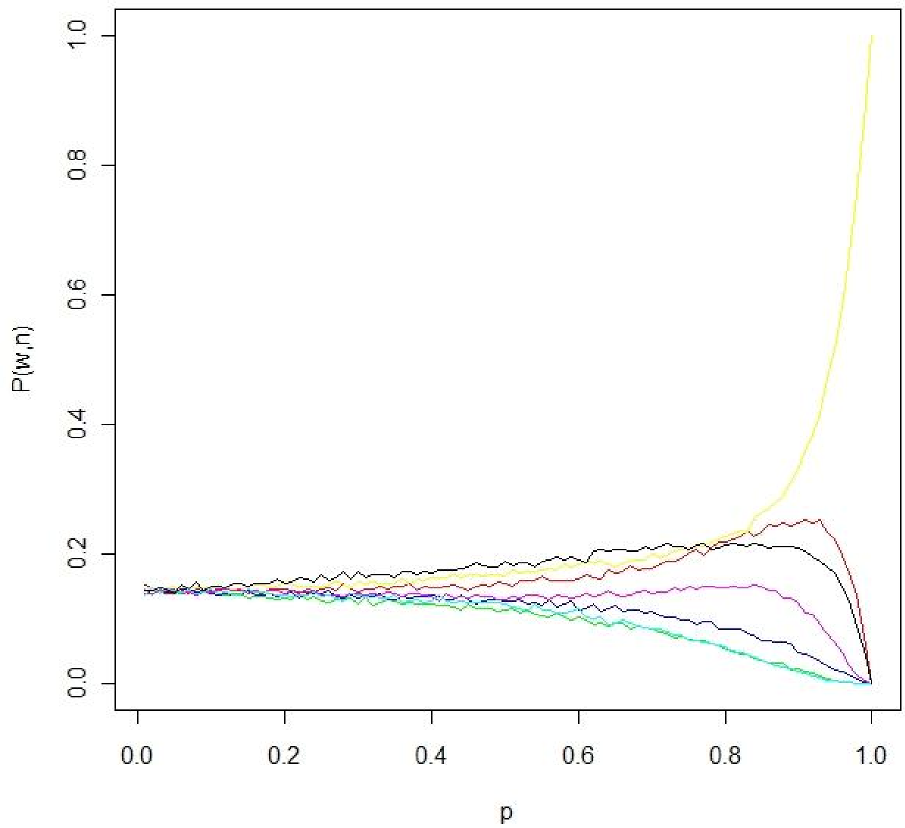
As you can see, the results are extremely close. Although we seem to have conquered Russian Roulette with the inclusion of the s parameter, we must rely solely on simulation to achieve arbitrary $P(w;n,k,s,p)$. Dr. Carlton and I were unable to generalize our matrix recursion to the case where $s\neq 1$. For this reason, when $s\neq 1$ and $k\neq 1$ we look to Mathematica for estimates of the actual probabilities.

***n* = 6, *k* = 4, *s* = 2**



For the sake of time, I merely changed s from 1 to 2. However, computing time aside, the sensitivity of the game to these types of small changes is worth investigating. For each $p = .05, .1, \ldots, 1$, 10,000 games were simulated and the relative frequencies calculated. Keep in mind that the winner of the game with $n = 6$, $k = 4$, $s = 2$ and $p = 1$ is the third player (blue on the above graph) instead of the fifth player in the $s = 1$ case. Once again it is nice to be able to observe seemingly intuitive things play out in the graph here. As in the $s = 1$ case we're seeing an uneven playing field from the start since $n = 6$ and $k = 4$ are not relatively prime. The second, fourth and sixth players start out with and maintain a lower probability of winning. On the other hand, the first, third and fifth players maintain higher probabilities. Just as we expect, the probability of the third player winning approaches 1 as p approaches 1 (the winner of the deterministic game).

$n = 7$, $k = 3$, $s = 3$



It took about 45 minutes to simulate the data displayed in the graph above. This makes sense though. Since $n = 7$ and $k = 3$ are relatively prime and I increased s to 3 shots it might take longer to simulate 1,000,000 games (10,000 for each value of $p = .01, .02, \ldots, 1$). None of the players in this scenario have a clear disadvantage because of their position, n, or k. This is further confirmed by the common starting probability for every player on the left side of the graph at $\frac{1}{7}$. As p goes to 1, the probability of the fourth player (the winner of the deterministic game) goes to 1.

To conclude this section, observe that this third variant of the game grants us one more interesting piece of information. The number of steps it takes to complete a game or determine a winner (the duration) follows a negative binomial distribution. We are in a sense playing until there have been $n - 1$ successful shots. Thus have the following expression for the expected duration of a game with n people, skip number k, probability of being eliminated p, and 1 shot to be eliminated. We can also look at the variance. Keep in mind that this is only true for the $s = 1$ case. Once s is greater than one, we can't be sure how many shots there will be before $n - 1$ players have been eliminated.

$$E(\text{duration}) = \frac{n-1}{p} \qquad Var(\text{duration}) = \frac{n-1}{p^2}$$

If $s>1$, then we can write the following knowing it takes s shots to eliminate a player and ns shots to eliminate every player.

$$\frac{s(n-1)}{p} \leq E(\text{duration}) \leq \frac{ns-1}{p}$$

3 Conclusion

To conclude this paper I would like to acknowledge the tools and ideas most useful in discovering the things we did about these different variations of the Josephus Problem. Key ideas in probability, linear algebra and number theory helped us re-establish and improve things that had already been shown for the classic scenario. We were able to utilize these subject areas along with Mathematica to find new and exciting results about both the probabilistic and multiple wounds variants of the original problem. We developed a useful recursion for finding any given $P(w;n,k,p)$. We can use simulation to verify our results and to estimate any given $P(w;n,k,p,s)$. Though we achieved much, I am still curious about new and different ways to expand this problem or analyze untouched parts of what we have already done.

4 Appendix

Below is code from Mathematica used to do everything described above that couldn't be done by just typing in a formula.

```
Deadly[total_, skip_, shots_, p_, rules___] :=
     Module[{ringarray, i, circ, temp, s, turnarray, j, shotarray, winner,
        woundarray, k, m, deadarray, all, lucky},
       winner = False;
       circ = Range[total];
       deadarray = Range[total];
       woundarray = Table[0, {m, 1, total - 1}];
       turnarray = Table[0, {j, 1, total}];
       shotarray = Table[0, {k, 1, total}];
       ringarray = Table[0, {i, 1, shots*total + 1}];
       ringarray[[1]] = circ;
       j = 1;
       While[winner == False,
         temp = {};
```

```
If[Mod[skip, Length[ringarray[[j]]]] != 0,
  s = Mod[skip, Length[ringarray[[j]]]],
   s = Length[ringarray[[j]]];
   ];
 If[woundarray[[-1]] != 0, AppendTo[woundarray, 0]];
 If[Length[ringarray[[j]]] >= skip,
   turnarray[[ringarray[[j]][[skip]]]] += 1;
   If[Floor[Random[] + p] == 1,
     shotarray[[ringarray[[j]][[skip]]]] += 1;

    woundarray[[Apply[Plus, shotarray]]] = ringarray[[j]][[skip]];
     ];
   If[shotarray[[ringarray[[j]][[skip]]]] == shots,
     temp = Delete[ringarray[[j]], skip];
     deadarray[[total - Length[temp]]] = ringarray[[j]][[skip]];,
     temp = ringarray[[j]];
     ];
   If[Length[ringarray] == j, AppendTo[ringarray, 0];];

  ringarray[[j + 1]] =
    RotateRight[Flatten[{temp}], Length[ringarray[[j]]] - skip];,
   If[Length[ringarray[[j]]] < skip,
       turnarray[[ringarray[[j]][[s]]]] += 1;
       If[Floor[Random[] + p] == 1,
         shotarray[[ringarray[[j]][[s]]]] += 1;

        woundarray[[Apply[Plus, shotarray]]] = ringarray[[j]][[s]];
         ];
       If[shotarray[[ringarray[[j]][[s]]]] == shots,
         temp = Delete[ringarray[[j]], s];

        deadarray[[total - Length[temp]]] = ringarray[[j]][[s]];,
         temp = ringarray[[j]];
         ];
       If[Length[ringarray] == j, AppendTo[ringarray, 0];];

      ringarray[[j + 1]] =
        RotateRight[Flatten[{temp}], Length[ringarray[[j]]] - s];
       ];
   ];
 j += 1;
```

```
    If[Length[ringarray[[j]]] == 1, winner = True;
      deadarray[[-1]] = ringarray[[j]][[1]];];
     ];
   all = everything /. {rules} /. {everything -> False};
   If[all == True,
     Print["Winner: ", ringarray[[j]]];
     Print["Turns: ", turnarray];
     Print["Shots: ", shotarray];
     Print["Wounded ", woundarray];
     Print["Dead ", deadarray];
     ];
   lucky = survivor /. {rules} /. {survivor -> True};
   If[lucky == True, Return[ringarray[[j]]];];
   (*Return[{ringarray[[j]], shotarray, turnarray}];*)
   ];

StochJosephus[total_, skip_, p_, rules___] :=
     Module[{q, Iarray, Rotarray, i, j, k, m, n, answarray, solarray,
        deterwin, gensol, crossarray, genwin, smgensol, crossing, ord, desc,
        descpos},
       If[total == 1, Return[{1}],
          q = 1 - p;
          Iarray = Table[IdentityMatrix[i], {i, 1, total}];
          Rotarray = Table[IdentityMatrix[m], {m, 1, total}];
          For[i = 1, i <= total, i++,
            For[j = 1, j <= i, j++,
                Rotarray[[i, j]] = RotateLeft[Rotarray[[i, j]], skip];
                ];
            ];
          matarray = Table[0, {n, 1, total}];
          answarray = Table[0, {m, 1, total}];
          solarray = Table[0, {j, 1, total}];

          matarray[[2]] = Iarray[[2]] + (-q)*Rotarray[[2]];
          If[EvenQ[skip], answarray[[2]] = {p, 0},
            If[OddQ[skip], answarray[[2]] = {0, p}]];
          solarray[[2]] = Inverse[matarray[[2]]].answarray[[2]];
          For[k = 3, k <= total, k++,
            matarray[[k]] = Iarray[[k]] + (-q)*Rotarray[[k]];

           answarray[[k]] = p*Rotarray[[k]].(Append[solarray[[k - 1]], 0]);
            solarray[[k]] = Inverse[matarray[[k]]].answarray[[k]];
            ];
```

```
(*deterwin = Deadly[total, skip];
  crossing = intersect /. {rules} /. {intersect -> False};
  If[crossing == True,
    gensol = StochJosephus[total, skip, b];
    genwin = gensol[[deterwin[[1]]]];
    crossarray = Table[0, {j, 1, total - 1}];
    smgensol = Delete[gensol, deterwin[[1]]];
    For[i = 1, i <= total - 1, i++,
      crossarray[[i]] = Solve[genwin == smgensol[[i]], b];
      ];
    Print["Winner: ", deterwin];
    Print[crossarray];
    ];
  *)
picture = graph /. {rules} /. {graph -> False};
If[picture == True,
  colorarray = Table[0, {i, 1, 7}];
  colorarray[[1]] = RGBColor[1, 0, 0];
  colorarray[[2]] = RGBColor[0, 1, 0];
  colorarray[[3]] = RGBColor[0, 0, 1];
  colorarray[[4]] = RGBColor[1, 1, 0];
  colorarray[[5]] = RGBColor[1, 0, 1];
  colorarray[[6]] = RGBColor[0, 1, 1];
  colorarray[[7]] = RGBColor[0, 0, 0];
  pics = Table[0, {k, 1, total}];
  lists = Table[Table[0, {m, 1, 100}], {n, 1, total}];
  r = 1;
  For[prob = .01, prob <= 1, prob += .01,
    game = StochJosephus[total, skip, prob];
    For[col = 1, col <= total, col++,
      lists[[col, r]] = {prob, game[[col]]};
      ];
    r++;
    ];
  For[plyr = 1, plyr <= total, plyr++,

     pics[[plyr]] =
         Graphics[{colorarray[[plyr]],
             Line[lists[[plyr]]]}, {Axes -> True,
             AxesLabel -> {"p", "P(w,n)"},
             DisplayFunction -> Identity}];
    ];
  Show[pics, DisplayFunction -> $DisplayFunction];
```

```
        ];
      ord = order /. {rules} /. {order -> False};
      If[ord == True,
        descpos = Range[total];
        desc = Sort[solarray[[total]], Greater];
        For[j = 1, j <= total, j++,

         descpos[[j]] =
             Flatten[Position[solarray[[total]], desc[[j]]]];];
        descpos = Flatten[descpos];
        Print[descpos];
        Print[desc];
        ];
      greatest = maximum /. {rules} /. {maximum -> False};
      If[greatest == True,

       Print["Greatest Probability: ", Max[solarray[[total]]],
           " at position ",
           Flatten[
             Position[solarray[[total]], Max[solarray[[total]]]]]];,
        Return[solarray[[total]]];
        ];

    ];
];
```

References

[1] Bogomolny, Alexander. "Josephus Flavius Problem Recursive Solution." *Cut-The-Knot*. 1996-2009. <http://www.cut-the-knot.org/recurrence/r_ solution.shtml>.

[2] Gunnar Blom; Jan-Eric Englund; Dennis Sandell. "General Russian Roulette." *Mathematics Magazine*, Vol. 69, No. 4 (Oct., 1996), pp. 293-297.