# Bowser's Big Blast in Mario Party 2

Diego Ortega Hernandez

Jan 2019

Luck based mini-games in the Mario Party series have been a bane of existence for some players – there's just no skill involved. Here, we prove why and decided to explore some probabilities and run simulations of Bowser's Big Blast.

## 1 Notation

- P1 is player 1, the same goes for P2, P3, etc.

## 2 Description of Bowser's Big Blast

This mini-game from Mario Party 2 starts with four players and five switches. There is exactly one switch that sets off a bomb. Each player sequentially picks their switch (so no player can select the same switch). As soon as a player selects a bomb, they are eliminated and the number of switches reduces.The bomb also randomly changes position as well.

E.g. Consider the following setup of switches, [B, S, S, S, S], where B is a bomb and S is safe. There are four players P1, P2, P3, P4. If they select

$$P1 \to 3$$
$$P2 \to 2$$
$$P3 \to 5$$
$$P4 \to 1$$

P4 is eliminated, the number of switches goes down to 4, the switches are shuffled, and the game continues.

## 3 Calculating the outcomes

### 3.1 Assumptions:

We assume (since two players cannot pick the same switch in the same round) that there must be a winner, and the players pick the switches without replacement.

### 3.2 Two players

We claim each player has the same probability of winning regardless of the round they play. Since the same situation happens with 4 players and 5 five switches as 2 players and 3 switches, we'll calculate the likelihood of outcomes for the latter.

We can break down the player's choices in the three scenarios. Observe the first two of these are stop conditions, and the last one is a loop condition that can result in the other two.

- P1 picks a bomb, and P2 is the winner.
- P1 does not pick a bomb, P2 picks a bomb, and P1 is the winner.

- P1 does not pick a bomb, P2 does not pick a bomb, then P1 picks another switch.

One can see a diagram of these situations and the accompanying probabilities in Figure 1.

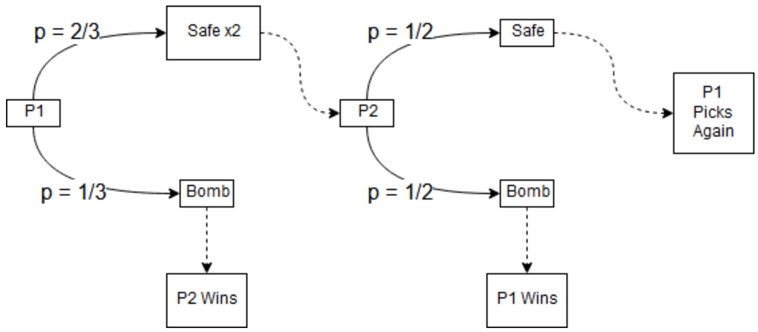


Figure 1: Tree diagram with probabilities for two players

The probability that P1 makes it to the next round is

$$P\left(P1 \text{ does not select a bomb}\right) = \frac{\#\text{ safe switches}}{\text{Total }\#\text{ of switches}} = \frac{2}{3}$$

We can deduce from this the probability that P2 wins

$$\begin{aligned} P\left(P2 \text{ wins}\right) &= P\left(P1 \text{ selects a bomb}\right) \\ &= 1 - P\left(P1 \text{ does not select a bomb}\right) \\ &= 1 - \frac{2}{3} \\ &= \frac{1}{3} \end{aligned}$$

The probability that P2 does not select a bomb is conditioned on the P1's choice since there is no replacement. So

$$P\left(P2 \text{ does not select a bomb} \mid P1 \text{ did not select a bomb}\right) = \frac{1}{2}$$

which we can also see since P1 removes one switch.

If P2 does select a bomb (again, conditioned on P1's choice), then the probability that P1 wins is (by Bayes' formula)

$$\begin{aligned} P\left(P1 \text{ wins}\right) &= P\left(P1 \text{ did not select a bomb} \cap P2 \text{ selects a bomb}\right) \\ &= P\left(P1 \text{ does not select a bomb}\right) \cdot P\left(P2 \text{ selects a bomb} \mid P1 \text{ did not select a bomb}\right) \\ &= P\left(P1 \text{ does not select a bomb}\right) \cdot \left[1 - P\left(P2 \text{ does not select a bomb} \mid P1 \text{ did not select a bomb}\right)\right] \\ &= \frac{2}{3} \cdot \frac{1}{2} \\ &= \frac{1}{3} \end{aligned}$$

Finally, we will calculate the probabilities for the both players to go again. By Bayes' formula again

$$\begin{aligned} P\left(P1 \text{ goes again next round}\right) &= P\left(P1 \text{ does not select a bomb} \cap P2 \text{ does not select a bomb}\right) \\ &= P\left(P1 \text{ does not select a bomb}\right) \cdot P\left(P2 \text{ does not select a bomb} \mid P1 \text{ did not select a bomb}\right) \\ &= \frac{2}{3} \cdot \frac{1}{2} \\ &= \frac{1}{3} \end{aligned}$$

$$
\begin{aligned}
P\left(P2 \text{ goes again next round}\right) &= P\left(\text{Both players make it to the next round}\right) \\
&= P\left(P1 \text{ did not select a bomb}\right) \cdot P\left(P2 \text{ does not select a bomb} \mid P1 \text{ did not select a bomb}\right) \\
&= \frac{2}{3} \cdot \frac{1}{2} \\
&= \frac{1}{3}
\end{aligned}
$$

So all outcomes are equally likely to be $\frac{1}{3}$ for two players.

### 3.3 Four players

We'll extend the idea for two players and create a larger diagram to describe the gameplay for four players in Figure 2. We omit the calculations, since they are setup exactly the same.

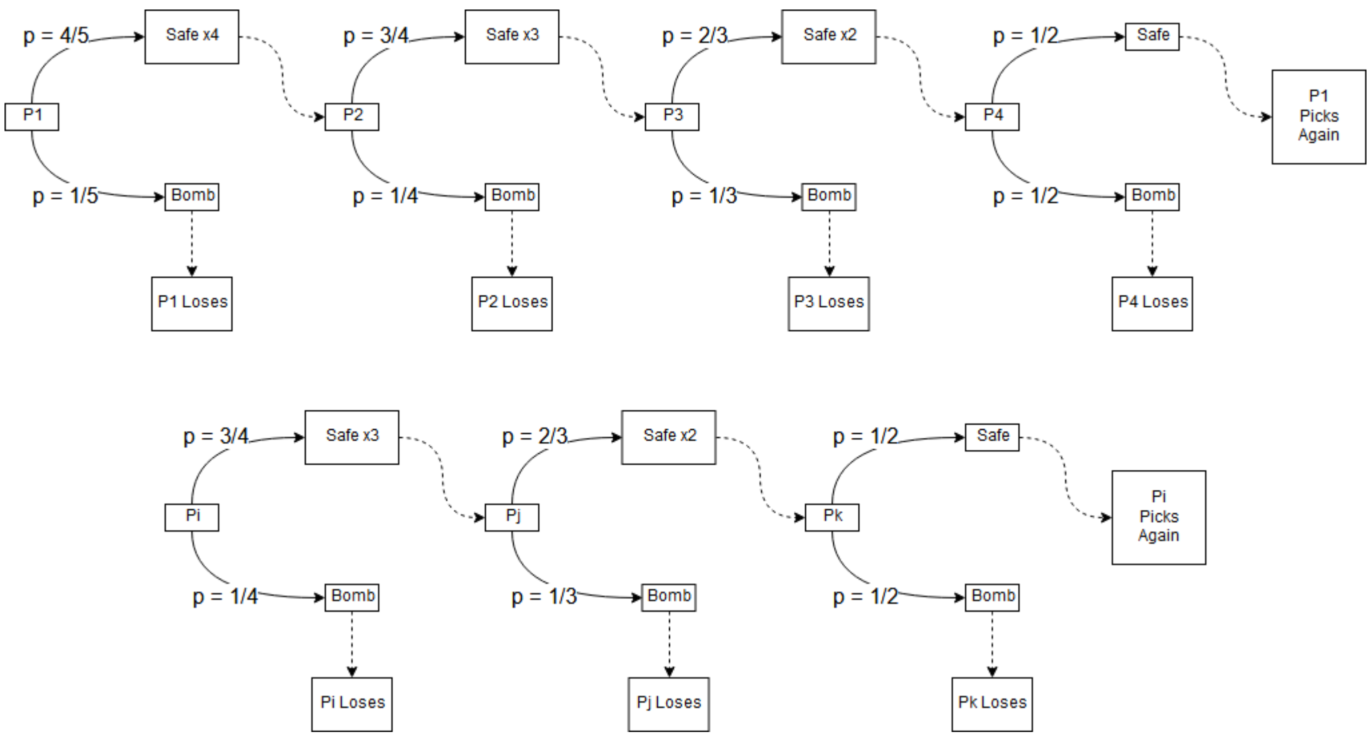


Figure 2: Tree diagram with probabilities for four and three players

From this, we can see all outcomes are equally likely for four players with each outcome having probability $\frac{1}{5}$, and $\frac{1}{4}$ for three players.

## 4 Expected number of rounds of play

TO be continued

## 5 Simulations

We compiled a simulation in Python 3.7 to imitate the gameplay in Bowser's Big Blast in order to calculate the expected number of rounds. We keep the same assumptions, except we decided to let all four players choose one at a time before they moved onto the next round.

On average? We got about 4 rounds.

Here's the full code.

```
import random
import numpy

#simulates one game of Bowser's Big Blast from Mario Party 2
#Assumes all players are equally likely to get a bomb.
#1 is the bomb
L = [0]*4 +[1]
player = [0,0,0,0]
count = 0
while len(L) > 2:
    #shuffles the list
    random.shuffle(L)
    #players 1-4 pick a random number (in order)
    choice = random.sample(L,(len(L)-1))
    count = count + 1
        #counts the number of players who are left
    if len(L) == 5:
        player[3] = player[3] + 1
    if len(L) == 4:
        player[2] = player[2] + 1
    if len(L) == 3:
        player[1] = player[1] + 1
         print(L)
    #As soon as one's pick is a bomb, they lose
    if choice.count(1) == 1:
        #the number of choices is reduced
        del L[L.index(0)] print(count) print(player)

################################

#Calculates the expectation of the number of rounds in each game starting with four players
#record all of the counts
Counts = []
Players = []
#1 million may be too much for some computers.
num_iter = 100000
for i in range(num_iter):
    #1 is the bomb
     L = [0]*4 +[1]
    player = [0,0,0,0]
    count = 0
    while len(L) > 2:
        #shuffles the list
        random.shuffle(L)
        #players 1-4 pick a random number (in order)
        choice = random.sample(L,(len(L)-1))
        count = count + 1
        #As soon as one's pick is a bomb, they lose
```

```
        if len(L) == 5:
            player[3] = player[3] + 1
        if len(L) == 4:
            player[2] = player[2] + 1
        if len(L) == 3:
            player[1] = player[1] + 1
        #if your pick is the bomb
        if choice.count(1) == 1:
            #the number of choices is reduced
             del L[L.index(0)]
    Counts.append(count)
    Players.append(player)

#Prints the expected value
print(sum(Counts)/num_iter)

#prints the average number of rounds with 4,3,2 players left
S = numpy.array([sum(i) for i in zip(*Players)]) print(S/num_iter)

##################################

print(max(Counts))

##################################

print(min(Counts))

##################################

#displays the scenarios where the number of rounds hits above a specified value
games = 0
for i in range(len(Players)):
    if max(Players[i]) > 6:
        print(Players[i])
        games = games + 1 print(games)
```

## References


[1] https://www.reddit.com/r/askmath/comments/63qpd8/solving_bowsers_big_blast_what_are_the_chances_of/

[2] https://www.reddit.com/r/statistics/comments/4np1lr/question_regarding_odds_in_a_mario_party_2/