Simulating Monopoly

Written by W. H. Bell

Monopoly is a very popular board game that involves strategy and chance. The board and the other components of the game are shown in Figure 1. Two dice are rolled to decide where a player will move to next, where the board has a total of 40 squares. The rules of moving around the board imply that a player is most likely to start their turn from the jail square. The combination of this effect with the most probable outcome of rolling two dice means that a player is more likely to land on some of the properties than the others. The goal of this project is to find the most likely streets by creating a simulation of the Monopoly board game.

Figure 1: the Monopoly game, produced by Hasbro

1) Rolling dice

An ideal six-sided die has an equal probability of landing on each of the six sides. The total probability for a system is defined as one. Therefore, the probability of landing on each of the sides is 1/6, since they all have an equal share of the total probability.

$$P(1)=P(2)=P(3)=P(4)=P(5)=P(6)=\frac{1}{6}$$

where ***P(n)*** is the probability of rolling ***n*** on a single die.

When two dice are rolled, the probability of one role is independent of the next. Therefore, the probability of rolling two sixes on two dice is:

$$P(6)\times P(6)=\frac{1}{36}$$

The probability of rolling a least one six on two dice is the sum of the probability of rolling six on each of the two dice,

$$P(6)+P(6)=\frac{2}{6}=\frac{1}{3}$$

In a monopoly game, the player moves according to the total value on the two dice. For example, a player might move five spaces forward if the player rolls one of the combinations { 2, 3 }, { 3, 2 }, {1, 4}, {4, 1}, where the first number corresponds to the first die and the second number corresponds to the second die. The probability of rolling each one of these combinations is the same:

$$P\{2,3\}=P(2)\times P(3)=\frac{1}{36}$$

$$P\{3,2\}=P(3)\times P(2)=\frac{1}{36}$$

$$P\{1,4\}=P(1)\times P(4)=\frac{1}{36}$$

$$P\{4,1\}=P(4)\times P(1)=\frac{1}{36}$$

The total probably of rolling five is the sum of each of probabilities for rolling one of these four combinations:

$$P\{2,3\}+P\{3,2\}+P\{1,4\}+P\{4,1\}=\frac{4}{36}=\frac{1}{9}$$

With dice, the number of combinations dictates the most probable outcome.

Write down the number of ways of making six, seven or eight on two dice. For example, for six there are { 1, 5 }, { 2, 4 }, { 3, 3 }...

_____

_____

_____

Out of six, seven or eight, which has the most combinations?

_____

Calculate the probabilities for rolling six, seven and eight on two dice.

_____

2) Simulating dice

Read through the `twoDice.py` program and try to understand each part of it. Then run the program by typing:

```
./twoDice.py
```

This program uses a random number generator instead of dice. When the random number generator is called it returns a value between one and six, with an equal probability of each of the numbers. This behaves in a similar manner as an ideal six-sided die. The function `rollTwoDice` calls the random number generator twice, sums the values of the two calls and returns the sum. In the main part of the program, the `rollTwoDice` function is called 100 times. Each of the resulting values is counted and put into a list. Then the total counts are converted into a probability by dividing by the total number of rolls. According to the program, which value is most likely?

_____

Try increasing the number of rolls from 100 to 1000000. What happens to the probability values?

_____

When the number of rolls is low, the statistical fluctuations will be more visible in the output of the program. When the number of rolls is large, the statistical fluctuations will have a smaller effect on the calculated probability values. When the number of rolls is large, compare the calculated probability for rolling five on two dice with the predicted value of 1/9. Compare the other values with your expectation of the probabilities of rolling six, seven and eight on two dice.

---

3) Simulating the board

It is often not necessary to simulate a system perfectly to predict something about the system. A simple simulator might only include the most important effects. Then more accurate simulations may include a few of the less important effects too. The benefit of producing simple simulations is that they can be written quickly and allow some crude judgements to be made. The complexity of a simulation is often a trade off between time spent producing the simulation and accuracy of the simulation.

Before adding anything else, the first thing to simulate about the Monopoly board is the squares on the board. The board squares can be numbered from 0 to 39. A player starts on the square numbered as zero, which is the “GO” square. The player then rolls two dice and goes around the board to the next square. The player continues to roll two dice until the game ends.

Read through the program `boardOnly.py`. Then try running the program by typing:

```
./boardOnly.py
```

This program will take a while to run. Therefore, you need to wait a moment for the answer. Does any question appear to be more likely than any other square?

_____

When you have finished running the program, close the histogram window before rerunning the program.

4) Adding “GO TO JAIL!”

The “GO TO JAIL!” square causes the player to go directly to jail. This means that if the player lands on the 30th square, then they have to go to the 10th square. The effect of this modification to the game is a small one, but can be seen by using a histogram to display each of the probability values. Read through the `goToJail.py` program. Then run it by typing:

```
./goToJail.py
```

This program will take a while to run. Therefore, you need to wait a moment for the answer. What has happened to the probability distribution?

In the Monopoly game, there are two decks of cards that also contain “GO TO JAIL!” commands and a card that commands the player to go to Pallmall. What is likely to happen to the probability distribution when these decks are added?

Compare your answer with the results of running

```
./twoDecksOfCards.py
```