Modified from: https://projecteuler.net/problem=84

In the game, *Monopoly*, the standard board is set up in the following way:

GO	A1	CC1	A2	T1	R1	B1	CH1	B2	B3	JAIL
H2										C1
T2										U1
H1										C2
CH3										C3
R4										R2
G3										D1
CC3										CC2
G2										D2
G1										D3
G2J	F3	U2	F2	F1	R3	E3	E2	CH2	E1	FP

A player starts on the GO square and adds the scores on two 6-sided dice to determine the number of squares they advance in a clockwise direction. Without any further rules we would expect to visit each square with equal probability: 2.5%. However, landing on G2J (Go To Jail), CC (community chest), and CH (chance) changes this distribution.

In addition to G2J, and one card from each of CC and CH, that orders the player to go directly to jail, if a player rolls three consecutive doubles, they do not advance the result of their 3rd roll. Instead they proceed directly to jail.

At the beginning of the game, the CC and CH cards are shuffled. When a player lands on CC or CH they take a card from the top of the respective pile and, after following the instructions, it is returned to the bottom of the pile. There are sixteen cards in each pile, but for the purpose of this problem we are only concerned with cards that order a movement; any instruction not concerned with movement will be ignored and the player will remain on the CC/CH square.

- Community Chest (2/16 cards):
    1. Advance to GO
    2. Go to JAIL
- Chance (10/16 cards):
    1. Advance to GO
    2. Go to JAIL
    3. Go back 3 squares.

We shall make no distinction between "Just Visiting" and being sent to JAIL, and we shall also ignore the rule about requiring a double to "get out of jail", assuming that they pay to get out on their next turn.

Assume that players start at GO and numbering the squares sequentially from 0 to 39.

Things you might need to use:

Rolling two six sided dice: RandomInteger[6,2] produces 2 draws from the numbers 1-6 and returns the results as a list of two numbers, i.e. {2,6} would represent one die coming up 2 and the other coming up 6.

A useful function will be Mod[x , k] which returns the remainder of division of x by k. We can use this to determine where on the board we are after moving from rolling the dice, because Mod[x, 40] will return x if it is 39 or less and will return the value having gone past Go if x is greater than 39.