Analyzing Monopoly

Jack Wu

11/23/2016

Monopoly Board game simulation

Rules for movement

The Monopoly Board is effectively a circle with 40 spaces on which a player can land. Players move from space to space around the board in a circle (square).

The number of spaces a player moves is determined by the roll of 2 dice. Most often, the player will roll the dice, land on a space, and end his turn there. If this were the entire game, the spaces would have a uniform distribution.

There are, however, several exceptions which provide the primary source of variation in space landing

Go to Jail

One space, "Go to Jail" sends players directly to jail (there is a jail space on the board). This space never counts as having been 'landed upon.' As soon as the player 'lands' here, he is immediately sent to jail, and the jail space gets counted as landed upon. This is the only space on the game board that moves a player's piece. The count of how often this space is landed on will always be 0.

Rolling Doubles

If a player rolls doubles (two of the same number), the player moves his piece, and then gets to roll the dice again for another move. However, if a player rolls doubles three times in a row, he is sent directly to jail. (The third space that the player would have 'landed on' does not count, but the jail space gets counted as landed on.)

Card Decks: Chance and Community Chest

A player can land on a "Chance" or "Community Chest" space. When a player lands on these spaces, he draws a card from the respective deck and follows its instructions. The instructions will sometimes give money to or take money from the player with no change in the player's position on the board. Other times, the card will instruct the player to move to another space on the board. The list of cards that can be drawn from each deck is provided below.

There are nine cards in the Chance deck that move the player's token. There are two cards in the Community Chest deck that move the player's token. All other cards do not move the player's token. For the sake of this simulation, you only need to worry about the cards that move the tokens.

A card may say 'move to the nearest railroad' or 'move to the nearest utility' or even 'go to property xxx'. In these cases, the player always moves forward. So if a player is on 'Oriental Avenue,' the nearest railroad is 'Pennsylvania Railroad' and NOT 'Reading Railroad.'

The Chance and Community Chest spaces always get counted as "landed on" even if the card drawn moves the player to another space or sends him to jail. In those cases, a tally is counted for the Chance/Community Chest space, the token is moved, and then a tally is counted for the space where the player ends his turn.

Jail

Jail is the most complicated aspect of this simulation.

If a player lands on space 11 (Jail), he is not in Jail. He is 'just visiting.' His play continues on as normal.

A player can be placed in jail in several ways: he rolls doubles three times in a row; he lands on the "go to jail" space; he draws a card that sends hims to jail.

When in jail, the player has the option to pay a fee to 'get out,' or he can choose not to pay the fee. If he pays the fee, he is out of jail, and his play continues normally as before. If he chooses not to pay the fee, he rolls the dice. If he rolls doubles on the dice, he gets out of jail and moves the number of spaces the dice show. However, even though he rolled doubles, he does NOT roll again. He takes his move out of jail and his turn ends. If he does not roll doubles, he stays in jail.

A player cannot stay in jail for more than three turns. On his third turn in jail, he rolls the dice and moves the number of spaces the dice show no matter what. If they are doubles, he moves those spaces for free. If he does not roll doubles, he moves those spaces, but must also pay a fee.

Play then continues as normal.

For this simulation, each time a player ends his turn in Jail, a tally will be counted as having been 'landed upon.'

I will simulate a 'long stay' strategy for Jail. This effectively means that the player will never choose to pay the fee to get out jail unless forced to do so. Effectively, this means that he will roll the dice and only leave jail if he gets doubles or it is his third turn in jail.

The Simulation

I will run 1,000 simulations of a two-player game that lasts 150 turns. This is a total of over 6 hundred thousand dice rolls - 1000 games x 150 turns x 2 players x 2 dice + additional rolls if the player gets doubles.

The Results

```r
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.2.5
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
gameboard <- data.frame(space = 1:40, title = c("Go" , "Mediterranean Avenue" , "Community Chest" , "Bal
chancedeck <- data.frame(index = 1:15, card = c("Advance to Go" , "Advance to Illinois Ave." , "Advance
communitydeck <- data.frame(index = 1:16, card = c("Advance to Go" , "Go to Jail" , "Bank error in your

## Write your code here

dice <- function(verbose=FALSE){
  faces <- sample(1:6, 2, replace=TRUE)
  if(faces[1] == faces[2]) doubles = TRUE
  else doubles = FALSE
  movement = sum(faces)
  if(verbose) cat("Rolled:", faces[1], faces[2], "\n")
  return(list(faces=faces, doubles=doubles, movement=movement))
}

player <- setRefClass("player",
  fields = list(
    pos = "numeric",      # position on the board
    verbose = "logical",
    jail = "numeric",
    rolldouble = "numeric"
  ),
  methods = list(
    move_n = function(n) {
      if(verbose) cat("Player at:", pos)
      if(verbose) cat(" Player moves:", n)
      pos <<- pos + n
      if(pos > 40) pos <<- pos - 40
      if(verbose) cat(" Player now at:", pos,"\n")
    },
    go_2_space_n = function(n){
      if(verbose) cat("Player at:", pos,".")
      pos <<- n
      if(verbose) cat(" Player now at:", pos,".\n")
    },
    go_2_jail = function(){
      if(verbose) cat("Player at:", pos,".")
      pos <<- 11
      if(verbose) cat(" Player now in jail.\n")
      jail <<- 1
    },
    increase_jail = function(){
      jail <<- jail + 1
    },
    reset_jail = function(){
      jail <<- 0
    },
    increase_double = function(){
      rolldouble <<- rolldouble + 1
    },
    reset_double = function(){
      rolldouble <<- 0
    }
  )
)

drawchance <- function(player, tracking,verbose = FALSE){
  draw <- sample(1:15, 1)
  #tally at chance if you draw a card that moves you
  if (draw %in% 1:9){
    tracking$increase_count(player$pos)
    if (verbose){
      cat("Tally at", player$pos, as.character(gameboard$title[player$pos]), "\n")
    }
  }
  if (verbose){
    cat("Drew Chance card", draw, "-", as.character(chancedeck$card[draw]),'\n')
  }
  if (draw == 1){
    player$go_2_space_n(1)
    if (verbose){
      cat("Player now at position 1:", as.character(gameboard$title[1]), "\n")
    }
  }
  if (draw == 2){
    player$go_2_space_n(25)
    if (verbose){
      cat("Player now at position 25:", as.character(gameboard$title[25]), "\n")
    }
  }
  if (draw == 3){
    player$go_2_space_n(12)
    if (verbose){
      cat("Player now at position 12:", as.character(gameboard$title[12]), "\n")
    }
  }
  if (draw == 4){
    if (player$pos == 8){
      player$go_2_space_n(13)
      if (verbose){
        cat("Player now at position 13:", as.character(gameboard$title[13]), "\n")
      }
    }
    if (player$pos == 23){
      player$go_2_space_n(29)
      if (verbose){
        cat("Player now at position 29:", as.character(gameboard$title[29]), "\n")
      }
    }
    if (player$pos == 37){
      player$go_2_space_n(5)
      if (verbose){
        cat("Player now at position 5:", as.character(gameboard$title[5]), "\n")
      }
    }
  }
  if (draw == 5){
    if (player$pos == 8){
      player$go_2_space_n(16)
      if (verbose){
        cat("Player now at position 16:", as.character(gameboard$title[16]), "\n")
      }
    }
    if (player$pos == 23){
      player$go_2_space_n(26)
      if (verbose){
        cat("Player now at position 26:", as.character(gameboard$title[26]), "\n")
      }
    }
    if (player$pos == 37){
      player$go_2_space_n(6)
      if (verbose){
        cat("Player now at position 6:", as.character(gameboard$title[6]), "\n")
      }
    }
  }
  if (draw == 6){
    player$go_2_space_n(6)
    if (verbose){
      cat("Player now at position 6:", as.character(gameboard$title[6]), "\n")
    }
  }
  if (draw == 7){
    player$go_2_space_n(40)
    if (verbose){
      cat("Player now at position 40:", as.character(gameboard$title[40]), "\n")
    }
  }
  if (draw == 8){
    player$go_2_jail()
    if (verbose){
      cat("Player now in Jail.\n")
    }
  }
  if (draw == 9){
    player$move_n(-3)
    if (verbose){
      cat("Player now at position", player$pos, as.character(gameboard$title[player$pos]), "\n")
    }
  }
}

drawcommunity <- function(player, tracking, verbose=FALSE){
  draw <- sample(1:16, 1)
  if (verbose){
    cat("Drew Community card", draw, "-", as.character(communitydeck$card[draw]),'\n')
  }
  #tally at chance if you draw a card that moves you
  if (draw %in% 1:2){
    tracking$increase_count(player$pos)
    if (verbose){
      cat("Tally at", player$pos, as.character(gameboard$title[player$pos]), "\n")
    }
  }
  if (draw == 1){
    player$go_2_space_n(1)
    if (verbose){
      cat("Player now at position 1:", as.character(gameboard$title[1]), "\n")
    }
  }
  if (draw == 2){
    player$go_2_jail()
    if (verbose){
      cat("Player now in Jail.\n")
    }
  }
}

# Space Tracking Reference Class ------------------------------------------

# a *basic* reference class to keep track of where people landed
tracking <- setRefClass("tracking",
  fields = list(
    tally = "numeric",
    verbose = "logical"
  ),
  methods = list(
    increase_count = function(n){
      tally[n] <<- tally[n] + 1
      if(verbose){
        cat("Tally at", n , ":", as.character(gameboard$title[n]), "\n")
      }
    }
  )
)

# Taking a turn -----------------------------------------------------------
# THIS IS THE MAIN FUNCTION!!
taketurn <- function(player, tracking, verbose=FALSE){
  roll <- dice()
  #keep track of whether or not player will roll again
  go_again <- roll$doubles
  #get out of jail if a double is rolled
  if (roll$doubles & player$jail){
    player$reset_jail()
    player$move_n(roll$movement)
    player$reset_double()
    go_again <- FALSE
  } else if (roll$doubles & !player$jail){
    player$increase_double()
    #if player rolled 3 doubles in a row, go to jail
    if (player$rolldouble == 3){
      player$go_2_jail()
      player$reset_double()
      go_again <- FALSE
    } else {
      player$move_n(roll$movement)
    }
  } else if (!roll$doubles & player$jail){
    #get out of jail if 3rd turn in jail
    if (player$jail == 3){
      player$move_n(roll$movement)
      player$reset_jail()
    } else {
      player$increase_jail()
    }
  } else if (!roll$doubles & !player$jail){
    #normal scenario
    player$move_n(roll$movement)
    player$reset_double()
  }
  #draw chance card
  if (player$pos %in% c(8,23,37)){
    drawchance(player, tracking, verbose)
    if (player$jail){
      go_again <- FALSE
    }
  }
  #draw community card
  if (player$pos %in% c(3,18,34)){
    drawcommunity(player, tracking, verbose)
    if (player$jail){
      go_again <- FALSE
    }
  }
  #if player isn't on "go to jail" spot
  if (player$pos != 31){
    #tally up
    tracking$increase_count(player$pos)
    #if player rolled a double, go again
    if (go_again){
      taketurn(player, tracking)
    }
  } else {
    #go to jail and tally up
    player$go_2_jail()
    tracking$increase_count(player$pos)
  }
}

set.seed(1)

space_tracking <- tracking$new(tally = rep(0,40), verbose = FALSE)
for(i in 1:1000){ # simulate 100 games
  #cat("#### NEW GAME",i,"##### \n")
  player1 <- player$new(pos = 1, jail = 0, rolldouble = 0, verbose = FALSE)  # create new players
  player2 <- player$new(pos = 1, jail = 0, rolldouble = 0, verbose = FALSE)
  for(i in 1:150){ # 150 turns for each game
    if(player1$verbose) cat("Player 1 turn\n")
    taketurn(player1, space_tracking)
    if(player2$verbose) cat("Player 2 turn\n")
    taketurn(player2, space_tracking)
  }
}
cat("1000 Games Ran \n")
```

```
## 1000 Games Ran
```

```
# the results after 100 turns. No rules have been implemented
results <- cbind(gameboard, tally = space_tracking$tally)
results <- cbind(results, rel = results$tally/sum(results$tally))
print(results)
```

```
##    space                 title tally        rel
## 1      1                    Go 10259 0.02788977
## 2      2  Mediterranean Avenue  6899 0.01875539
## 3      3       Community Chest  7049 0.01916317
## 4      4         Baltic Avenue  7064 0.01920395
## 5      5            Income Tax  8317 0.02261031
## 6      6      Reading Railroad  9562 0.02599493
## 7      7       Oriental Avenue  7559 0.02054964
## 8      8                Chance  7925 0.02154463
## 9      9        Vermont Avenue  7933 0.02156638
## 10    10    Connecticut Avenue  7846 0.02132987
## 11    11                  Jail 40444 0.10994968
## 12    12     St. Charles Place  9228 0.02508693
## 13    13      Electric Company  8925 0.02426320
## 14    14         States Avenue  7554 0.02053605
## 15    15       Virginia Avenue  8607 0.02339870
## 16    16 Pennsylvania Railroad  8901 0.02419796
## 17    17       St. James Place  9438 0.02565782
## 18    18       Community Chest  9288 0.02525004
## 19    19      Tennessee Avenue  9870 0.02683225
## 20    20       New York Avenue  9872 0.02683768
## 21    21          Free Parking  9633 0.02618795
## 22    22       Kentucky Avenue  9112 0.02477157
## 23    23                Chance  9696 0.02635921
## 24    24        Indiana Avenue  9131 0.02482323
## 25    25       Illinois Avenue 10430 0.02835464
## 26    26        B & O Railroad  9635 0.02619338
## 27    27       Atlantic Avenue  8761 0.02381736
## 28    28        Ventnor Avenue  8776 0.02385813
## 29    29           Water Works  9329 0.02536150
## 30    30        Marvin Gardens  8268 0.02247710
## 31    31            Go to jail     0 0.00000000
## 32    32        Pacific Avenue  8770 0.02384182
## 33    33 North Carolina Avenue  8605 0.02339326
## 34    34       Community Chest  8936 0.02429310
## 35    35   Pennsylvania Avenue  8032 0.02183552
## 36    36   Short Line Railroad  7745 0.02105529
## 37    37                Chance  7546 0.02051430
## 38    38            Park Place  7139 0.01940784
## 39    39            Luxury Tax  7000 0.01902996
## 40    40             Boardwalk  8757 0.02380648
```

```
sum(results$tally)
```

```
## [1] 367841
```

```
arrange(results, desc(rel))
```

```
##    space                 title tally        rel
## 1     11                  Jail 40444 0.10994968
## 2     25       Illinois Avenue 10430 0.02835464
## 3      1                    Go 10259 0.02788977
## 4     20       New York Avenue  9872 0.02683768
## 5     19      Tennessee Avenue  9870 0.02683225
## 6     23                Chance  9696 0.02635921
## 7     26        B & O Railroad  9635 0.02619338
## 8     21          Free Parking  9633 0.02618795
## 9      6      Reading Railroad  9562 0.02599493
## 10    17       St. James Place  9438 0.02565782
## 11    29           Water Works  9329 0.02536150
## 12    18       Community Chest  9288 0.02525004
## 13    12     St. Charles Place  9228 0.02508693
## 14    24        Indiana Avenue  9131 0.02482323
## 15    22       Kentucky Avenue  9112 0.02477157
## 16    34       Community Chest  8936 0.02429310
## 17    13      Electric Company  8925 0.02426320
## 18    16 Pennsylvania Railroad  8901 0.02419796
## 19    28        Ventnor Avenue  8776 0.02385813
## 20    32        Pacific Avenue  8770 0.02384182
## 21    27       Atlantic Avenue  8761 0.02381736
## 22    40             Boardwalk  8757 0.02380648
## 23    15       Virginia Avenue  8607 0.02339870
## 24    33 North Carolina Avenue  8605 0.02339326
## 25     5            Income Tax  8317 0.02261031
## 26    30        Marvin Gardens  8268 0.02247710
## 27    35   Pennsylvania Avenue  8032 0.02183552
## 28     9        Vermont Avenue  7933 0.02156638
## 29     8                Chance  7925 0.02154463
## 30    10    Connecticut Avenue  7846 0.02132987
## 31    36   Short Line Railroad  7745 0.02105529
## 32     7       Oriental Avenue  7559 0.02054964
## 33    14         States Avenue  7554 0.02053605
## 34    37                Chance  7546 0.02051430
## 35    38            Park Place  7139 0.01940784
## 36     4         Baltic Avenue  7064 0.01920395
## 37     3       Community Chest  7049 0.01916317
## 38    39            Luxury Tax  7000 0.01902996
## 39     2  Mediterranean Avenue  6899 0.01875539
## 40    31            Go to jail     0 0.00000000
```