Final\_q5

Amber Duevel

4/26/2022

#checking to see if you roll a double  
check\_double<-function(face, double\_count){  
 if(face[1]==face[2]){  
 double\_count=double\_count+1  
 if(double\_count==3){  
 player\_df[player+1, 2] = 11  
 assign("player\_df",player\_df,.GlobalEnv)  
 return(sum(face))  
 }  
 else {  
 space = player\_df[player+1, 2] + sum(face)  
 if (space == 40) {  
 space = 40  
 }  
 else {  
 space = space%%40  
 }  
   
 mono\_board[space, 2] = as.numeric(mono\_board[space, 2] + 1)  
 player\_df[player+1, 2] = space  
   
 mono\_graph\_data = rbind(mono\_graph\_data, c(player\_df[player+1, 1], mono\_board[player\_df[player+1, 2], 3], mono\_board[player\_df[player+1, 2], 4]))  
   
 face = sample(1:6,2,replace=TRUE)  
 check\_double(face, double\_count)  
 }  
 }else{  
 #return the sum of the faces and go back to dice function  
 assign("mono\_board", mono\_board, .GlobalEnv)  
 assign("player\_df", player\_df, .GlobalEnv)  
 assign("mono\_graph\_data", mono\_graph\_data, .GlobalEnv)  
 return(sum(face))  
 }  
}  
  
#call this function if the player is in jail  
jail\_out<-function(face, jail\_count){  
 if(face[1]==face[2]){  
 movement=sum(face)  
 }else{  
 jail\_count=jail\_count+1  
 #leave jail if player has already tried to get out of jail three times and was unsuccessful   
 if(jail\_count==3){  
 movement=sum(face)  
 }  
 return(jail\_count)  
 }  
}  
#add more movement things to this function instead of check\_double to let it be more organized.  
dice <- function(){  
 double\_count = 0  
 jail\_count = 0  
 face = sample(1:6,2,replace=TRUE)  
   
 if(player\_df[player+1, 2] == 11){  
 while(jail\_count<3){  
 jail\_out(face, jail\_count)  
 break  
 }  
 }  
   
 sum\_face = check\_double(face, double\_count)  
 return(sum\_face)  
}

library(dplyr)

##   
## 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

#creating the community chess cards  
cc\_cards = data.frame()  
cc1 = c("Advance to Go(collect $200)", 200, TRUE, "Go")  
cc2 = c("Bank error in your favor. Collect $200", 200, FALSE, NA)  
cc3 = c("Doctor's fee. Pay $50", -50, FALSE, NA)  
cc4 = c("From sale of stock you get $50", 50, FALSE, NA)  
cc5 = c("Get out of jail free", 0, FALSE, NA)  
cc6 = c("Go to jail. Go directly to jail, do not pass Go, do not collect $200", 0, TRUE, "Just Visiting")  
cc7 = c("Holiday fund matures. Receive $100", 100, FALSE, NA)  
cc8 = c("Income tax refund. Collect $20", 20, FALSE, NA)  
cc9 = c("It is your birthday. Collect $10 from every player", 0, FALSE, NA)  
cc10 = c("Life insurance matures. Collect $100", 100, FALSE, NA)  
cc11 = c("Pay hospital fees of $100", -100, FALSE, NA)  
cc12 = c("Pay school fees of $50", -50, FALSE, NA)  
cc13 = c("Receive $25 consultancy fee", -25, FALSE, NA)  
cc14 = c("You are assessed for street repair. $40 per house. $115 per hotel", 0, FALSE, NA)  
cc15 = c("You have won second prize in a beauty contest. Collect $10", 10, FALSE, NA)  
cc16 = c("You inherit $100", 100, FALSE, NA)  
cc\_cards <- rbind(cc\_cards, cc1, cc2, cc3, cc4, cc5, cc6, cc7, cc8, cc9, cc10, cc11, cc12, cc13, cc14, cc15, cc16)  
colnames(cc\_cards) = c("card\_description", "money\_gain", "true/false", "move")  
cc\_cards = transform(cc\_cards, money\_gain = as.numeric(money\_gain))  
cc\_cards\_vec <- c("Advance to Go(collect $200)", "Bank error in your favor. Collect $200", "Doctor's fee. Pay $50", "From sale of stock you get $50", "Get out of jail free", "Go to jail. Go directly to jail, do not pass Go, do not collect $200", "Holiday fund matures. Receive $100", "Income tax refund. Collect $20", "It is your birthday. Collect $10 from every player", "Life insurance matures. Collect $100", "Pay hospital fees of $100", "Pay school fees of $50", "Receive $25 consultancy fee", "You are assessed for street repair. $40 per house. $115 per hotel", "You have won second prize in a beauty contest. Collect $10", "You inherit $100")  
  
#selecting a community chess card  
jail\_cc = function(space) {  
 mono\_board[space, 2] = as.numeric(mono\_board[space, 2] + 1)  
 player\_df[player+1, 2] = space  
 mono\_graph\_data = rbind(mono\_graph\_data, c(player\_df[player+1, 1], mono\_board[player\_df[player+1, 2], 3], mono\_board[player\_df[player+1, 2], 4]))  
 assign("player\_df",player\_df,.GlobalEnv)  
   
 r <- sample(cc\_cards\_vec, size = 1, replace = FALSE)  
 cat("CC...", r)  
 space = player\_df[player+1, 2]  
 if (r == "Go to jail. Go directly to jail, do not pass Go, do not collect $200") {  
 space = 11  
 }  
 else if(r == "Advance to Go(collect $200)") {  
 space = 1  
 }else{  
 space=space  
 }  
 r\_pos = match(r, cc\_cards[,1])  
 assign("player\_df",player\_df,.GlobalEnv)  
 assign("mono\_board", mono\_board, .GlobalEnv)  
 assign("mono\_graph\_data", mono\_graph\_data, .GlobalEnv)  
 return(c(space, r\_pos))  
}  
  
#creating the chance cards  
chance\_cards = data.frame()  
c1 = c("Advance to Boardwalk", 0, TRUE, "Boardwalk")  
c2 = c("Advance to Go(collect $200)", 200, TRUE, "Go")  
c3 = c("Advance to Illinois Avenue. If you pass Go, collect $200", 0, TRUE, "Illinois Avenue")  
c4 = c("Advance to St. Charles Place. If you pass Go, collect $200", 0, TRUE, "St. Charles Place")  
c5 = c("Advance to the nearest Railroad. If unowned, you may buy it from the Bank. If owned, pay owner twice the rental to which they are otherwise entitled", 0, TRUE, "Railroad") #Needs to look for the nearest railroad, which it's not doing right now  
c6 = c("Advance to the nearest Railroad. If unowned, you may buy it from the Bank. If owned, pay owner twice the rental to which they are otherwise entitled", 0, TRUE, "Railroad") #Same as the previous one  
c7 = c("Advance token to the nearest Utility. If unowned, you may buy it from the Bank. If owned, throw dice and pay owner a total ten times amount thrown.", 0, TRUE, "Utility") #Same as the previous one but with utilities  
c8 = c("Bank pays you dividend of $50", 50, FALSE, NA)  
c9 = c("Get out of jail free", 0, FALSE, NA)  
c10 = c("Go back 3 spaces", 0, TRUE, "Current space minus 3")  
c11 = c("Go to jail. Go directly to jail, do not pass Go, do not collect $200", 0, TRUE, "Just Visiting")  
c12 = c("Make general repairs on all your property. For each house pay $25. For each hotel pay $100", 0, FALSE, NA)  
c13 = c("Speeding fine $15", -15, FALSE, NA)  
c14 = c("Take a trip to Reading Railroad. If you pass Go, collect $200", 0, TRUE, "Go")  
c15 = c("You have been elected Chairman of the Board. Pay each player $50", -50, FALSE, NA)  
c16 = c("Your building loan matures. Collect $150", 150, FALSE, NA)  
chance\_cards = rbind(chance\_cards, c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11, c12, c13, c14, c15, c16)  
colnames(chance\_cards) = c("card\_description", "money\_gain", "true/false", "move")  
chance\_cards = transform(chance\_cards, money\_gain = as.numeric(money\_gain))  
chance\_cards\_vec <- c("Advance to Boardwalk", "Advance to Go(collect $200)", "Advance to Illinois Avenue. If you pass Go, collect $200", "Advance to St. Charles Place. If you pass Go, collect $200", "Advance to the nearest Railroad. If unowned, you may buy it from the Bank. If owned, pay owner twice the rental to which they are otherwise entitled", "Advance to the nearest Railroad. If unowned, you may buy it from the Bank. If owned, pay owner twice the rental to which they are otherwise entitled", "Advance token to the nearest Utility. If unowned, you may buy it from the Bank. If owned, throw dice and pay owner a total ten times amount thrown.", "Bank pays you dividend of $50", "Get out of jail free", "Go back 3 spaces", "Go to jail. Go directly to jail, do not pass Go, do not collect $200", "Make general repairs on all your property. For each house pay $25. For each hotel pay $100", "Speeding fine $15", "Take a trip to Reading Railroad. If you pass Go, collect $200", "You have been elected Chairman of the Board. Pay each player $50", "Your building loan matures. Collect $150")  
  
#selecting a chance card  
jail\_c = function(space) {  
 mono\_board[space, 2] = as.numeric(mono\_board[space, 2] + 1)  
 player\_df[player+1, 2] = space  
 mono\_graph\_data = rbind(mono\_graph\_data, c(player\_df[player+1, 1], mono\_board[player\_df[player+1, 2], 3], mono\_board[player\_df[player+1, 2], 4]))  
   
 r1 <- sample(chance\_cards\_vec, size = 1, replace = FALSE)  
 print(r1)  
 space = player\_df[player+1, 2]  
 if (r1 == "Go to jail. Go directly to jail, do not pass Go, do not collect $200") {  
 space = 11  
 }  
 else if (r1 == "Advance to Boardwalk") {  
 space = 40  
 }  
 else if (r1 == "Advance to Go(collect $200)") {  
 space = 1  
 }  
 else if (r1 == "Advance to Illinois Avenue. If you pass Go, collect $200") {  
 space = 25  
 }  
 else if (r1 == "Advance to St. Charles Place. If you pass Go, collect $200") {  
 space = 12  
 }  
 else if (r1 == "Advance to the nearest Railroad. If unowned, you may buy it from the Bank. If owned, pay owner twice the rental to which they are otherwise entitled") {  
 if (space > 1 && space < 6) {  
 space = 6  
 }  
 else if (space < 16 && space > 6) {  
 space = 16  
 }  
 else if (space < 26 && space > 16) {  
 space = 26  
 }  
 else {  
 space = 36  
 }  
 }  
 else if (r1 == "Advance token to the nearest Utility. If unowned, you may buy it from the Bank. If owned, throw dice and pay owner a total ten times amount thrown.") {  
 if (space > 1 && space < 13) {  
 space = 13  
 }  
 else if (space < 29 && space > 13) {  
 space = 29  
 }  
 else {  
 space = 13  
 }  
 }  
 else if (r1 == "Go back 3 spaces") {  
 space = space - 3  
 if(space<1){  
 space=space+40  
 }  
 }  
 else if (r1 == "Take a trip to Reading Railroad. If you pass Go, collect $200") {  
 space = 6  
 }  
 else {  
 space=space  
 }  
 r\_pos = match(r1, chance\_cards[,1])  
 assign("player\_df",player\_df,.GlobalEnv)  
 assign("mono\_board", mono\_board, .GlobalEnv)  
 assign("mono\_graph\_data", mono\_graph\_data, .GlobalEnv)  
 return(c(space, r\_pos))  
}

#establishing to money amounts to use in the function  
bank\_money = 0  
parking\_money = 0  
  
#adding or subtracting the money to the players total amount  
money\_board<-function(){  
 #checking if player lands on free parking  
 if(player\_df[player+1, 2] == 21){  
 free\_parking()  
 bank\_money = 0  
 }  
 #if player is at GO add 200 to money amount  
 if(player\_df[player+1, 2] == 1){  
 player\_df[player+1,3]=player\_df[player+1,3]+200  
 }  
}  
  
#adding money to player amount based on the community chess cards  
cc\_money<-function(space\_pos){  
 r\_pos = space\_pos[2]  
 player\_df[player+1,3] = player\_df[player+1,3] + cc\_cards[r\_pos, 2]  
 if(cc\_cards[r\_pos, 2] < 0){  
 bank\_money = bank\_money - cc\_cards[r\_pos, 2]  
 assign("player\_df", player\_df, .GlobalEnv)  
 }  
 else {  
 assign("player\_df", player\_df, .GlobalEnv)  
 }  
}  
  
#adding money to players amount based on the chance cards  
chance\_money<-function(space\_pos){  
 r\_pos = space\_pos[2]  
 player\_df[player+1,3] = player\_df[player+1,3] + chance\_cards[r\_pos, 2]  
 if(chance\_cards[r\_pos, 2] < 0){  
 bank\_money = bank\_money - chance\_cards[r\_pos, 2]  
 assign("player\_df", player\_df, .GlobalEnv)  
 }  
 else {  
 assign("player\_df", player\_df, .GlobalEnv)  
 }  
}  
  
#adding money amount to free parking so there is total when someone lands on that spot  
free\_parking<-function(){  
 if(parking\_money == 0){  
 player\_df[player+1,3]=player\_df[player+1,3] + 100  
 assign("player\_df", player\_df, .GlobalEnv)  
 return(player\_df[player+1,3])  
 }  
 parking\_money = parking\_money + bank\_money  
 player\_df[player+1,3] = player\_df[player+1,3] + parking\_money  
}

mono\_board = data.frame()  
# All of the spaces on the board  
go = c("Go", 0, 200, 0, "None")  
mediterranean = c("Mediterranean Avenue", 0, 172, 0, "Purple")  
community1 = c("Community Chest1", 0, 155, 0, "Cards")  
baltic = c("Baltic Avenue", 0, 136, 0, "Purple")  
income = c("Income Tax", 0, 118, 0, "None")  
reading = c("Reading Railroad", 0, 100, 0, "Railroad")  
oriental = c("Oriental Avenue", 0, 82, 0, "Light Blue")  
chance1 = c("Chance1", 0, 64, 0, "Cards")  
vermont = c("Vermont Avenue", 0, 46, 0, "Light Blue")  
connecticut = c("Connecticut Avenue", 0, 28, 0, "Light Blue")  
just\_visiting\_jail = c("Just Visiting/Jail", 0, 0, 0, "None")  
charles = c("St. Charles Place", 0, 0, 28, "Pink")  
electric = c("Electric Company", 0, 0, 46, "Utilities")  
states = c("States Avenue", 0, 0, 64, "Pink")  
virginia = c("Virginia Avenue", 0, 0, 82, "Pink")  
penn\_railroad = c("Pennsylvania Railroad", 0, 0, 100, "Railroad")  
james = c("St. James Place", 0, 0, 118, "Orange")  
community2 = c("Community Chest2", 0, 0, 136, "Cards")  
tennessee = c("Tennessee Avenue", 0, 0, 154, "Orange")  
new\_york = c("New York Avenue", 0, 0, 172, "Orange")  
parking = c("Free Parking", 0, 0, 200, "None")  
kentucky = c("Kentucky Avenue", 0, 28, 200, "Red")  
chance2 = c("Chance2", 0, 46, 200, "Cards")  
indiana = c("Indiana Avenue", 0, 64, 200, "Red")  
illinois = c("Illinois Avenue", 0, 82, 200, "Red")  
b\_o = c("B&O Railroad", 0, 100, 200, "Railroad")  
atlantic = c("Atlantic Avenue", 0, 118, 200, "Yellow")  
ventnor = c("Ventnor Avenue", 0, 136, 200, "Yellow")  
water = c("Water Works", 0, 155, 200, "Utilities")  
marvin = c("Marvin Gardens", 0, 172, 200, "Yellow")  
jail = c("Go to Jail", 0, 200, 200, "None")  
pacific = c("Pacific Avenue", 0, 200, 172, "Green")  
carolina = c("North Carolina Avenue", 0, 200, 154, "Green")  
community3 = c("Community Chest3", 0, 200, 136, "Cards")  
penn\_avenue = c("Pennsylvania Avenue", 0, 200, 118, "Green")  
short = c("Short Line", 0, 200, 100, "Railroad")  
chance3 = c("Chance3", 0, 200, 82, "Cards")  
park = c("Park Place", 0, 200, 64, "Dark Blue")  
luxury = c("Luxury Tax", 0, 200, 46, "None")  
boardwalk = c("Boardwalk", 0, 200, 28, "Dark Blue")  
mono\_board = rbind(mono\_board, go, mediterranean, community1, baltic, income, reading, oriental, chance1, vermont, connecticut, just\_visiting\_jail, charles, electric, states, virginia, penn\_railroad, james, community2, tennessee, new\_york, parking, kentucky, chance2, indiana, illinois, b\_o, atlantic, ventnor, water, marvin, jail, pacific, carolina, community3, penn\_avenue, short, chance3, park, luxury, boardwalk)  
colnames(mono\_board) = c('space\_name', 'count', 'x\_dimension', 'y\_dimension', 'space\_color')

mono\_board = transform(mono\_board, x\_dimension = as.numeric(x\_dimension))  
mono\_board = transform(mono\_board, y\_dimension = as.numeric(y\_dimension))  
mono\_board = transform(mono\_board, count = as.numeric(count))

#creating a data frame with all the players variables  
player\_df = data.frame()  
player1 = c("player1", 1, 1500)  
player2 = c("player2", 1, 1500)  
player3 = c("player3", 1, 1500)  
player4 = c("player4", 1, 1500)  
player\_df = rbind(player\_df, player1, player2, player3, player4)  
colnames(player\_df) = c("name", "position", "money")  
player\_df = transform(player\_df, position = as.numeric(position))  
player\_df = transform(player\_df, money = as.numeric(money))  
num\_players = nrow(player\_df)  
#Columns: player, x-coordinate, y-coordinate  
mono\_graph\_data = data.frame()  
mono\_graph\_data = rbind(c(player\_df[1, 1], mono\_board[player\_df[1, 2], 3], mono\_board[player\_df[1, 2], 4]),   
 c(player\_df[2, 1], mono\_board[player\_df[2, 2], 3], mono\_board[player\_df[2, 2], 4]),   
 c(player\_df[3, 1], mono\_board[player\_df[3, 2], 3], mono\_board[player\_df[3, 2], 4]),   
 c(player\_df[4, 1], mono\_board[player\_df[4, 2], 3], mono\_board[player\_df[4, 2], 4]))  
colnames(mono\_graph\_data) = c('player', 'x\_coordinate', 'y\_coordinate')  
rounds = 0  
player = 0  
  
#moving the players to different positions based on the roll  
move = function() {  
 roll = dice()  
 space = player\_df[player+1, 2] + roll  
 if (space == 40) {  
 space = 40  
 }  
 else {  
 space = space%%40  
 }  
 if(space==3 || space==18 || space==34){  
 space\_pos=jail\_cc(space)  
 cc\_money(space\_pos)  
 }else if(space==8 || space==23 || space==37){  
 space\_pos=jail\_c(space)  
 chance\_money(space\_pos)  
 }  
 money\_board()  
 assign("mono\_board", mono\_board, .GlobalEnv)  
 assign("player\_df", player\_df, .GlobalEnv)  
 assign("mono\_graph\_data", mono\_graph\_data, .GlobalEnv)  
 return(space)  
}  
  
#calling player turn so each player can move based on their roll  
player\_turn = function(player) {  
 s = move()  
 mono\_board[s, 2] = as.numeric(mono\_board[s, 2] + 1)  
 player\_df[player+1, 2] = s  
   
 mono\_graph\_data = rbind(mono\_graph\_data, c(player\_df[player+1, 1], mono\_board[player\_df[player+1, 2], 3], mono\_board[player\_df[player+1, 2], 4]))  
   
 return(c(assign("mono\_board", mono\_board, .GlobalEnv), assign("player\_df", player\_df, .GlobalEnv), assign("mono\_graph\_data", mono\_graph\_data, .GlobalEnv)))  
}  
  
#the amount of rounds we want to run  
while (rounds < 100) {  
 player\_turn(player)  
 cat(" Player: ", player\_df[player+1,1], " Position: ", mono\_board[player\_df[player+1, 2], 1], "\n")  
 player = (player+1)%%num\_players  
 rounds = rounds + 1  
}

## [1] "Go to jail. Go directly to jail, do not pass Go, do not collect $200"  
## Player: player1 Position: Chance1   
## Player: player2 Position: Reading Railroad   
## Player: player3 Position: Reading Railroad   
## Player: player4 Position: Vermont Avenue   
## Player: player1 Position: St. James Place   
## Player: player2 Position: States Avenue   
## Player: player3 Position: St. Charles Place   
## Player: player4 Position: States Avenue   
## Player: player1 Position: Indiana Avenue   
## [1] "Advance to Illinois Avenue. If you pass Go, collect $200"  
## Player: player2 Position: Chance2   
## Player: player3 Position: Tennessee Avenue   
## Player: player4 Position: Free Parking   
## Player: player1 Position: Go to Jail   
## Player: player2 Position: Pacific Avenue   
## Player: player3 Position: Ventnor Avenue   
## Player: player4 Position: Water Works   
## Player: player1 Position: Park Place   
## Player: player2 Position: Short Line   
## Player: player3 Position: Pennsylvania Avenue   
## Player: player4 Position: North Carolina Avenue   
## CC... Receive $25 consultancy fee Player: player1 Position: Community Chest1   
## Player: player2 Position: Baltic Avenue   
## Player: player3 Position: Baltic Avenue   
## Player: player4 Position: Luxury Tax   
## Player: player1 Position: Oriental Avenue   
## Player: player2 Position: Just Visiting/Jail   
## Player: player3 Position: Electric Company   
## Player: player4 Position: Income Tax   
## CC... Doctor's fee. Pay $50 Player: player1 Position: Community Chest2   
## CC... Pay school fees of $50 Player: player2 Position: Community Chest2   
## Player: player3 Position: Kentucky Avenue   
## [1] "Advance to Go(collect $200)"  
## Player: player4 Position: Chance1   
## Player: player1 Position: B&O Railroad   
## Player: player2 Position: B&O Railroad   
## Player: player3 Position: Marvin Gardens   
## Player: player4 Position: Electric Company   
## Player: player1 Position: Pacific Avenue   
## Player: player2 Position: Go to Jail   
## CC... Advance to Go(collect $200) Player: player3 Position: Community Chest3   
## Player: player4 Position: Pennsylvania Railroad   
## Player: player1 Position: Go   
## Player: player2 Position: Park Place   
## Player: player3 Position: Boardwalk   
## Player: player4 Position: Free Parking   
## [1] "Advance to St. Charles Place. If you pass Go, collect $200"  
## Player: player1 Position: Chance1   
## Player: player2 Position: Mediterranean Avenue   
## Player: player3 Position: Oriental Avenue   
## Player: player4 Position: Pacific Avenue   
## Player: player1 Position: Pennsylvania Railroad   
## Player: player2 Position: Connecticut Avenue   
## Player: player3 Position: Electric Company   
## CC... Go to jail. Go directly to jail, do not pass Go, do not collect $200 Player: player4 Position: Community Chest1   
## Player: player1 Position: Tennessee Avenue   
## Player: player2 Position: Pennsylvania Railroad   
## [1] "Advance to the nearest Railroad. If unowned, you may buy it from the Bank. If owned, pay owner twice the rental to which they are otherwise entitled"  
## Player: player3 Position: Chance2   
## Player: player4 Position: Electric Company   
## Player: player1 Position: Atlantic Avenue   
## [1] "Make general repairs on all your property. For each house pay $25. For each hotel pay $100"  
## Player: player2 Position: Chance2   
## Player: player3 Position: Marvin Gardens   
## Player: player4 Position: Indiana Avenue   
## Player: player1 Position: North Carolina Avenue   
## Player: player2 Position: Marvin Gardens   
## Player: player3 Position: Park Place   
## Player: player4 Position: Marvin Gardens   
## Player: player1 Position: Baltic Avenue   
## Player: player2 Position: Park Place   
## Player: player3 Position: Go   
## Player: player4 Position: Park Place   
## Player: player1 Position: States Avenue   
## CC... Receive $25 consultancy fee Player: player2 Position: Community Chest1   
## [1] "Advance to Boardwalk"  
## Player: player3 Position: Chance1   
## Player: player4 Position: Income Tax   
## Player: player1 Position: Tennessee Avenue   
## Player: player2 Position: Connecticut Avenue   
## Player: player3 Position: St. James Place   
## Player: player4 Position: States Avenue   
## Player: player1 Position: Marvin Gardens   
## Player: player2 Position: Pennsylvania Railroad   
## Player: player3 Position: B&O Railroad   
## Player: player4 Position: Indiana Avenue   
## Player: player1 Position: Park Place   
## [1] "Go back 3 spaces"  
## Player: player2 Position: Chance2   
## Player: player3 Position: North Carolina Avenue   
## Player: player4 Position: North Carolina Avenue   
## Player: player1 Position: Income Tax   
## Player: player2 Position: Pacific Avenue   
## Player: player3 Position: Luxury Tax   
## Player: player4 Position: Boardwalk   
## Player: player1 Position: St. Charles Place   
## Player: player2 Position: Luxury Tax   
## Player: player3 Position: Reading Railroad   
## Player: player4 Position: Oriental Avenue   
## Player: player1 Position: New York Avenue   
## Player: player2 Position: Income Tax   
## Player: player3 Position: St. Charles Place   
## Player: player4 Position: Connecticut Avenue   
## Player: player1 Position: B&O Railroad   
## Player: player2 Position: Connecticut Avenue   
## [1] "Advance to the nearest Railroad. If unowned, you may buy it from the Bank. If owned, pay owner twice the rental to which they are otherwise entitled"  
## Player: player3 Position: Chance2   
## CC... Holiday fund matures. Receive $100 Player: player4 Position: Community Chest2

library(grid)  
library(jpeg)  
library(tidyverse)

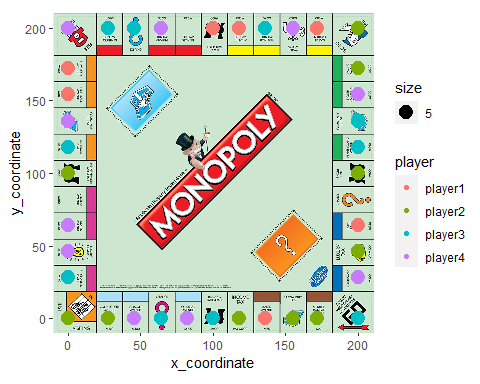
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.6 v stringr 1.4.0  
## v tidyr 1.2.0 v forcats 0.5.1  
## v readr 2.1.2

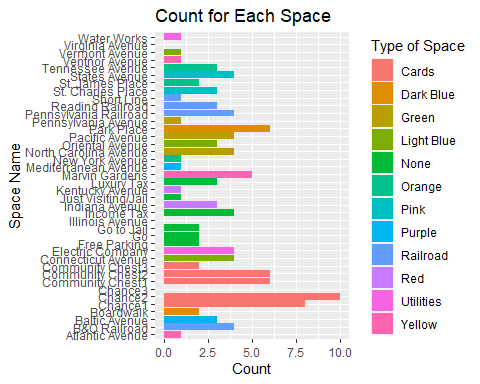
## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(ggplot2)  
library(gganimate)  
  
#creating the monopoly board to display on the board  
Sys.setenv("VROOM\_CONNECTION\_SIZE" = 131072 \* 2)  
board\_img = rasterGrob(readJPEG("monopolyboard.jpg"),   
 width = unit(1, "npc"), height = unit(1, "npc"))

#graphing the all the positions that each player lands on  
mono\_graph\_data = transform(mono\_graph\_data, x\_coordinate = as.numeric(x\_coordinate))  
mono\_graph\_data = transform(mono\_graph\_data, y\_coordinate = as.numeric(y\_coordinate))  
mono\_graph = ggplot(mono\_graph\_data, group = "player") +  
 annotation\_custom(board\_img) +  
 theme(aspect.ratio = 1) +  
 scale\_x\_continuous(limits = c(0, 200)) +  
 scale\_y\_continuous(limits = c(0, 200)) +  
 geom\_point(aes(x = x\_coordinate, y = y\_coordinate, color = player, size = 5))  
mono\_graph



#creating a bar graph with the count of each space getting landed on  
count\_plot = ggplot(mono\_board, aes(x = space\_name, y = count), group = "space\_color") +  
 geom\_bar(stat = "identity", aes(fill = space\_color)) +  
 coord\_flip() +  
 ggtitle("Count for Each Space") +  
 ylab("Count") +  
 xlab("Space Name") +  
 labs(fill = "Type of Space")  
count\_plot



With our program we don’t have anything dealing with properties. Due to this players usually don’t become bankrupt and that is when a normal game would. It is a rare occasion when a player loses money so it’s hard for a player to become bankrupt. This is why we only ran our program for one hundred rounds. We created a function to collect all the money taken away from each player and added it to bank amount. This would be given to a player if they landed on free parking. The more rounds we run the more likely the bank amount would increase significantly. This would help player’s money amount to also increase significantly too.