Final\_q4

Amber Duevel

4/26/2022

# Checking for rolled doubles and sending the player to jail if the faces are equal  
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))  
 }  
 # Otherwise assigning the new space to the old space plus the roll of the dice  
 else {  
 space = player\_df[player+1, 2] + sum(face)  
 if (space == 40) {  
 space = 40  
 }  
 else {  
 space = space%%40  
 }  
 # Conditions for drawing community chest and chance cards  
 if(space==3 || space==18 || space==34){  
 space=jail\_cc(space)  
 }else if(space==8 || space==23 || space==37){  
 space=jail\_c(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]))  
   
 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))  
 }  
}  
# Counting how many times the player has been in jail  
jail\_out<-function(face, jail\_count){  
 if(face[1]==face[2]){  
 movement=sum(face)  
 }  
 else {  
 jail\_count=jail\_count+1  
 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

# The deck of community chest 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\_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")  
# Checking if player movement is needed based on the cc card draw  
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]))  
   
 #sampling one card from the cc deck and checking the card description  
 r <- sample(cc\_cards\_vec, size = 1, replace = FALSE)  
 print(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  
 }  
 assign("player\_df",player\_df,.GlobalEnv)  
 return(space)   
}  
# The deck of 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(cc\_cards) = c("card\_description", "money\_gain", "true/false", "move")  
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")  
# Checking if player movement is needed based on the c card draw  
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]))  
   
 #selecting a random chance card  
 r1 <- sample(chance\_cards\_vec, size = 1, replace = FALSE)  
 print(r1)  
   
 #checking what the card is and moving the player to the position the card says  
 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  
 }  
 assign("player\_df",player\_df,.GlobalEnv)  
 return(space)  
}

#mono\_board is a dataframe which holds 40 space objects, with Go being the first space  
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')  
#using transform() to have the x-dimension, y-dimension, and count columns be numeric instead of character  
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))

#player\_df dataframe which is storing position and money data for each player  
player\_df = data.frame()  
#four players were created with position = 1 and starting money is 1500  
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))  
num\_players = nrow(player\_df)  
#mono\_graph\_data is a dataframe which stores the history of each players moves around the board  
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  
#move function which stores roll from dice() output and add roll to the current position in player\_df and stores in space, and space is returned  
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=jail\_cc(space)  
 }else if(space==8 || space==23 || space==37){  
 space=jail\_c(space)  
 }  
 return(space)  
}  
#player\_turn takes in the current player, calls the move function and stores the current space in s, then increases the count on that space by 1 and updates player\_df to be the new current space  
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)))  
}  
#running 100 rounds of the game  
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  
}

## Player: player1 Position: Connecticut Avenue   
## [1] "Advance to St. Charles Place. If you pass Go, collect $200"  
## Player: player2 Position: St. Charles Place   
## Player: player3 Position: Oriental Avenue   
## [1] "You have been elected Chairman of the Board. Pay each player $50"  
## Player: player4 Position: Chance1   
## Player: player1 Position: States Avenue   
## Player: player2 Position: St. James Place   
## Player: player3 Position: Pennsylvania Railroad   
## Player: player4 Position: Virginia Avenue   
## Player: player1 Position: Free Parking   
## Player: player2 Position: Ventnor Avenue   
## Player: player3 Position: Free Parking   
## [1] "Advance to Illinois Avenue. If you pass Go, collect $200"  
## Player: player4 Position: Illinois Avenue   
## Player: player1 Position: Atlantic Avenue   
## [1] "Income tax refund. Collect $20"  
## Player: player2 Position: Community Chest3   
## Player: player3 Position: Ventnor Avenue   
## Player: player4 Position: North Carolina 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: player1 Position: Boardwalk   
## Player: player2 Position: Luxury Tax   
## Player: player3 Position: Luxury Tax   
## Player: player4 Position: Mediterranean Avenue   
## Player: player1 Position: Income Tax   
## Player: player2 Position: Connecticut Avenue   
## Player: player3 Position: Reading Railroad   
## Player: player4 Position: Reading Railroad   
## Player: player1 Position: Just Visiting/Jail   
## Player: player2 Position: St. James Place   
## Player: player3 Position: Just Visiting/Jail   
## Player: player4 Position: St. Charles Place   
## Player: player1 Position: Tennessee Avenue   
## Player: player2 Position: Indiana Avenue   
## [1] "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."  
## Player: player3 Position: Water Works   
## Player: player4 Position: Tennessee Avenue   
## Player: player1 Position: Indiana Avenue   
## Player: player2 Position: North Carolina Avenue   
## Player: player3 Position: Pacific Avenue   
## Player: player4 Position: Indiana Avenue   
## Player: player1 Position: Go to Jail   
## Player: player2 Position: Luxury Tax   
## [1] "You have been elected Chairman of the Board. Pay each player $50"  
## Player: player3 Position: Chance3   
## Player: player4 Position: Marvin Gardens   
## [1] "Go to jail. Go directly to jail, do not pass Go, do not collect $200"  
## Player: player1 Position: Just Visiting/Jail   
## Player: player2 Position: Income Tax   
## Player: player3 Position: Income Tax   
## Player: player4 Position: Pennsylvania Avenue   
## Player: player1 Position: Tennessee Avenue   
## Player: player2 Position: St. Charles Place   
## Player: player3 Position: Connecticut Avenue   
## Player: player4 Position: Go   
## Player: player1 Position: Kentucky Avenue   
## Player: player2 Position: Tennessee Avenue   
## Player: player3 Position: St. James Place   
## Player: player4 Position: Vermont Avenue   
## Player: player1 Position: Pacific Avenue   
## Player: player2 Position: Indiana Avenue   
## Player: player3 Position: New York Avenue   
## Player: player4 Position: Virginia Avenue   
## Player: player1 Position: Luxury Tax   
## Player: player2 Position: North Carolina Avenue   
## Player: player3 Position: Indiana Avenue   
## Player: player4 Position: Kentucky Avenue   
## [1] "Speeding fine $15"  
## Player: player1 Position: Chance1   
## [1] "You inherit $100"  
## Player: player2 Position: Community Chest1   
## Player: player3 Position: North Carolina Avenue   
## Player: player4 Position: Ventnor Avenue   
## Player: player1 Position: Just Visiting/Jail   
## Player: player2 Position: St. Charles Place   
## [1] "Life insurance matures. Collect $100"  
## Player: player3 Position: Community Chest1   
## Player: player4 Position: Go to Jail   
## Player: player1 Position: Kentucky Avenue   
## [1] "Go to jail. Go directly to jail, do not pass Go, do not collect $200"  
## Player: player2 Position: Just Visiting/Jail   
## Player: player3 Position: Connecticut Avenue   
## Player: player4 Position: Short Line   
## Player: player1 Position: Ventnor Avenue   
## [1] "Holiday fund matures. Receive $100"  
## Player: player2 Position: Community Chest2   
## Player: player3 Position: Pennsylvania Railroad   
## [1] "From sale of stock you get $50"  
## Player: player4 Position: Community Chest1   
## Player: player1 Position: Luxury Tax   
## Player: player2 Position: Free Parking   
## Player: player3 Position: Indiana Avenue   
## Player: player4 Position: Just Visiting/Jail   
## Player: player1 Position: Income Tax   
## Player: player2 Position: Pacific Avenue   
## Player: player3 Position: Go to Jail   
## Player: player4 Position: Free Parking   
## Player: player1 Position: St. Charles Place   
## [1] "Get out of jail free"  
## Player: player2 Position: Chance3   
## Player: player3 Position: Short Line   
## Player: player4 Position: Ventnor Avenue   
## Player: player1 Position: Pennsylvania Railroad   
## [1] "Advance to Go(collect $200)"  
## Player: player2 Position: Go   
## [1] "Bank error in your favor. Collect $200"  
## Player: player3 Position: Community Chest1   
## Player: player4 Position: Go to Jail   
## [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: player1 Position: B&O Railroad   
## Player: player2 Position: Baltic 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: Pennsylvania Railroad   
## Player: player4 Position: Go   
## Player: player1 Position: North Carolina Avenue   
## Player: player2 Position: Connecticut Avenue   
## Player: player3 Position: Free Parking   
## Player: player4 Position: St. Charles Place

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

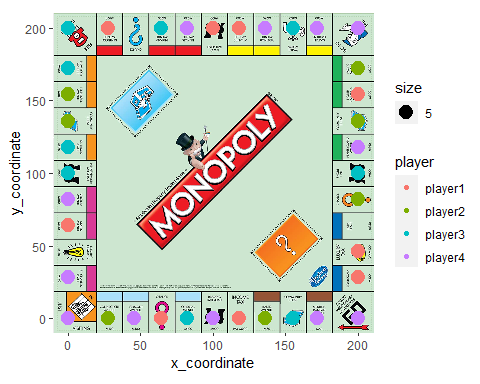
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## 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

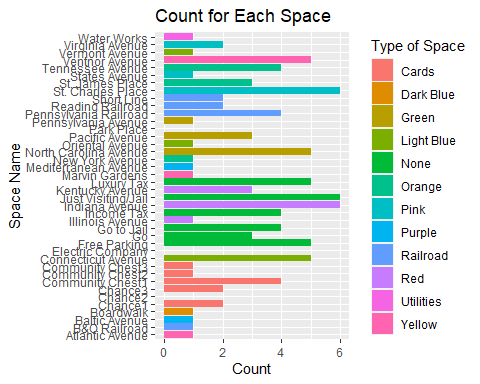
## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

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

#using the transform() function to change the x and y-coordinate columns to numeric in mono\_graph\_data  
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 is a scatterplot with the monopoly board as the background and the points are the plots where the players have been. There are four different colors, which represent each player  
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



#count\_plot is a bar graph with the space name as the y-axis and count for each space as the x-axis. Colors are grouped based on the type of space and color associated with each space on the board, if any  
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



Now that we added the movement cards into the game, the number of times each space is landed on is less equally distributed. The spaces that players can move to when drawing a card from either chance or community chest include Utilities, St. James Place, St. Charles Place, Railroads, Marvin Gardens, Luxury Tax, Boardwalk, and Go. These spaces should have a higher probability of being landed on compared to the rest of the spaces. This is different than a game where the cards are not being used because we are basing movement off of both random dice rolls and cards. When the game is only running for 100 rounds, the counts for each space do not differ that much and the differences cannot necessarily be explained by the cards. However, with more rounds or data from multiple games, you would see that the spaces that are associated with the cards would have a higher count than the rest of the spaces.