Battlefield – 2P81 Group project

Camden Macoretta
Abdullah Obeidi
Atheer Alqadri
Heduin Ravell Bandeira de Morais

Objective

The objective of the game is to have a higher sum of cards than the dealer.

Components

- One player and dealer (minimum)
- 52 card deck
- Cash value chips

General Rules

Cards 2-10 are equal to the value listed on each specific card. Face cards are equal to 10 and aces are equal to 11. No sum of cards can be less than 4 or greater than 22. The dealer must always deal counterclockwise. In the event of a tie, the players original cards become dead and are not included in the new sum. The maximum number of cards that can be dealt to a player during the game is four.

Setup

The dealer will begin by asking the player(s) to place their bets in cash value chips. After the initial bet is placed, it cannot be changed into a lower amount. The dealer will then deal each player one individual card face up while finishing with themselves. The player(s) will receive their second card at a later time.

Starting The Game

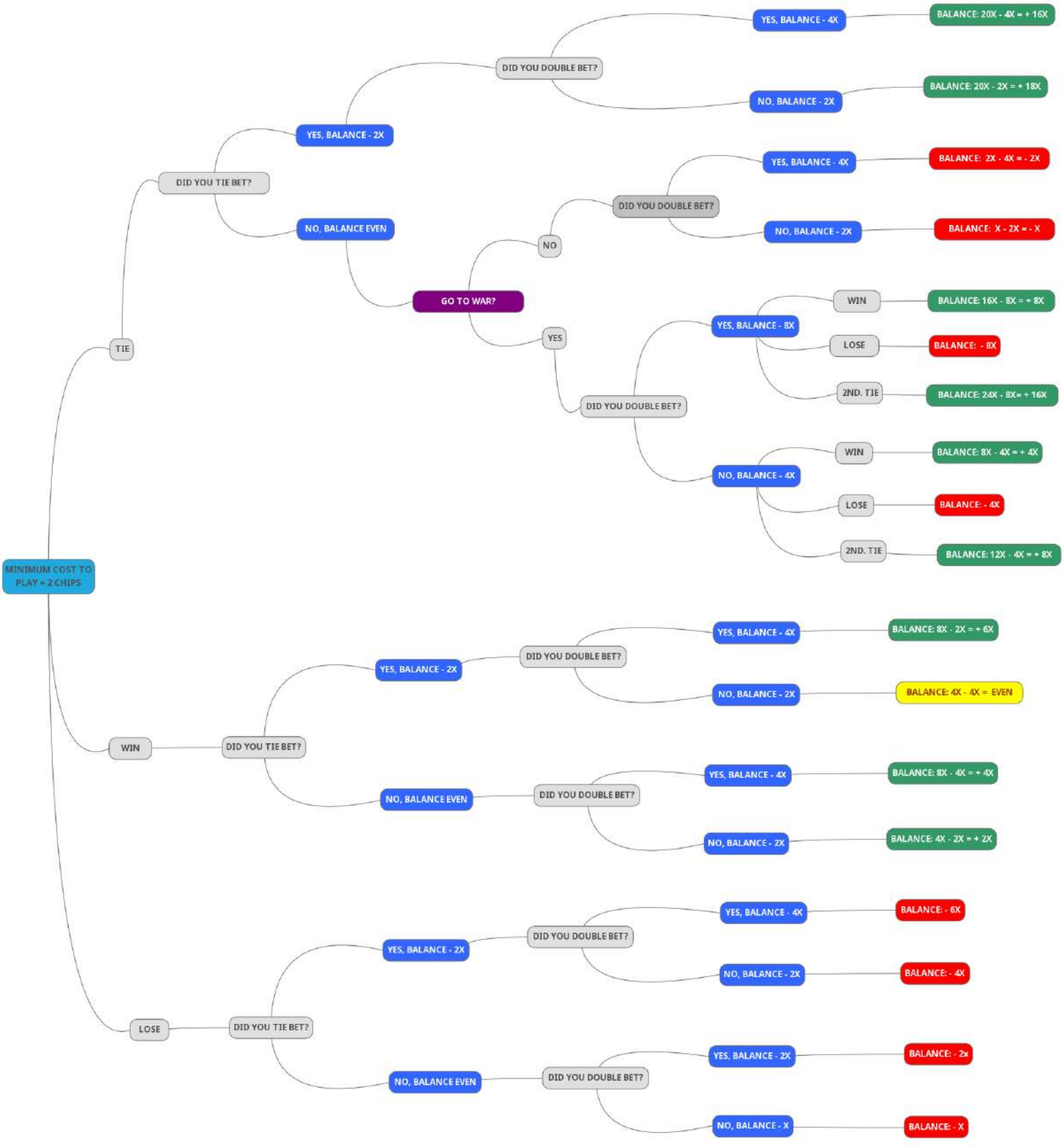
Once the player(s) received their first card they have three options. The player(s) can either keep their initial bet the same or double it. The player(s) can also place an independent side bet of the same value of their initial bet that their sum of cards will equal the sum of the dealer.

Winning and Losing The Game

After the second round of potential bets are placed, the dealer deals the player(s) their second card. If the dealer has a higher sum of cards than the player(s) then the dealer wins their total bets placed. If the dealer has a lower sum of cards than the player(s) then the player(s) wins twice fold their total bets placed. If the player(s) bet on a tie, they win tenfold their side bet. In the event of a tie in which the player(s) didn't bet on a tie, they will have the option to take half their bet and leave half for the dealer or double their total bets placed. They are no longer able to bet on a tie.

Second Turn

If the player(s) chooses to double after a tie, the dealer deals them two new cards face up and then themselves two new cards. The same rules apply if the dealer has a lower or higher sum than the player(s). In the event of another tie, the player wins threefold their total bets.

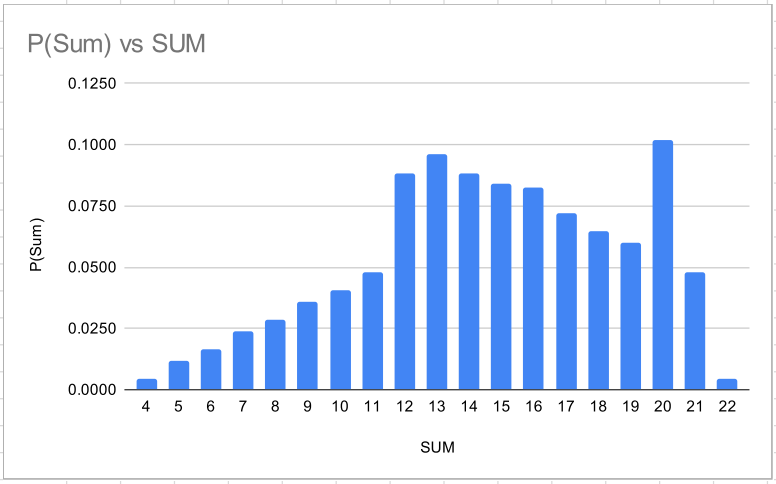

MINIMUM COST TO PLAY = 2 CHIPS
TIE
DID YOU TIE BET?
YES, BALANCE - 2X
DID YOU DOUBLE BET?
YES, BALANCE - 4X
BALANCE: 20X - 4X = + 16X
NO, BALANCE - 2X
BALANCE: 20X - 2X = + 18X
NO, BALANCE EVEN
GO TO WAR?
NO
DID YOU DOUBLE BET?
YES, BALANCE - 4X
BALANCE: 2X - 4X = - 2X
NO, BALANCE - 2X
BALANCE: X - 2X = - X
YES
DID YOU DOUBLE BET?
YES, BALANCE - 8X
WIN
BALANCE: 16X - 8X = + 8X
LOSE
BALANCE: - 8X
2ND. TIE
BALANCE: 24X - 8X= + 16X
NO, BALANCE - 4X
WIN
BALANCE: 8X - 4X = + 4X
LOSE
BALANCE: - 4X
2ND. TIE
BALANCE: 12X - 4X = + 8X
WIN
DID YOU TIE BET?
YES, BALANCE - 2X
DID YOU DOUBLE BET?
YES, BALANCE - 4X
BALANCE: 8X - 2X = + 6X
NO, BALANCE - 2X
BALANCE: 4X - 4X = EVEN
NO, BALANCE EVEN
DID YOU DOUBLE BET?
YES, BALANCE - 4X
BALANCE: 8X - 4X = + 4X
NO, BALANCE - 2X
BALANCE: 4X - 2X = + 2X
LOSE
DID YOU TIE BET?
YES, BALANCE - 2X
DID YOU DOUBLE BET?
YES, BALANCE - 4X
BALANCE: - 6X
NO, BALANCE - 2X
BALANCE: - 4X
NO, BALANCE EVEN
DID YOU DOUBLE BET?
YES, BALANCE - 2X
BALANCE: - 2x
NO, BALANCE - X
BALANCE: - X

Let's consider the following arrangement of possible pairs and sums of face values. Let's count how many there are.

Qtys	Cards	2	3	4	5	6	7	8	9	10	J	Q	K	A
12	**2**	**4**	5	6	7	8	9	10	11	12	12	12	12	13
32	**3**	5	**6**	7	8	9	10	11	12	13	13	13	13	14
44	**4**	6	7	**8**	9	10	11	12	13	14	14	14	14	15
64	**5**	7	8	9	**10**	11	12	13	14	15	15	15	15	16
76	**6**	8	9	10	11	**12**	13	14	15	16	16	16	16	17
96	**7**	9	10	11	12	13	**14**	15	16	17	17	17	17	18
108	**8**	10	11	12	13	14	15	**16**	17	18	18	18	18	19
128	**9**	11	12	13	14	15	16	17	**18**	19	19	19	19	20
236	**10**	12	13	14	15	16	17	18	19	**20**	20	20	20	21
-	**J**	12	13	14	15	16	17	18	19	20	**20**	20	20	21
-	**Q**	12	13	14	15	16	17	18	19	20	20	**20**	20	21
-	**K**	12	13	14	15	16	17	18	19	20	20	20	**20**	21
256	**A**	13	14	15	16	17	18	19	20	21	21	21	21	**22**
	Qtys	**256**	**236**	**224**	**220**	**192**	**172**	**160**	**272**	**128**	**-**	**-**	**-**	**12**

Examples:
For Qty sum 5, two ways to be dealt
2, 3 or 3, 2.
each with 4 possible ways.
so 2*4*4=32
For sum 6 (and all even sums)
there is one way (same face value),
with counts 4*3, when equal cards, so
`2*4*4+1*4*3 = 44`

SUM	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	19	***possible sums***
QTY	**12**	**32**	**44**	**64**	**76**	**96**	**108**	**128**	**236**	**256**	**236**	**224**	**220**	**192**	**172**	**160**	**272**	**128**	**12**	***2668***	***qty of each sum***
P(Sum)	0.0045	0.0120	0.0165	0.0240	0.0285	0.0360	0.0405	0.0480	0.0885	0.0960	0.0885	0.0840	0.0825	0.0720	0.0645	0.0600	0.1019	0.0480	0.0045	1.0000	total



`Let's compute PT(Sum) probability of a player's hand be involved in a TIE after the second card is dealt.`

First DECISION: BET on TIES or NOT

For Ties, possible sums are calculated based on initial cards from both player and dealer, as follows:

SET A: For the person with the lower face card, possible tie-sums goes from LowerCard + 2 up to LowerCard + 11

SET B: For the person with the higher face card, possible tie-sums go from HigherCard + 2 up to HigherCard + 11

SET A n SET B: The list of possible sums goes from HigherCard + 2 up to LowerCard + 11. See examples below:

Example 1: Player gets 4, Dealer gets 9. This example gives 5 possible sums: 11, 12, 13, 14, 15 for a tie-value.

This way, in the table the point (Player's 1st. Card, Dealer's 1st Card) = (4, 9) will be P(11)+P(12)+P(13)+P(14)+P(15)= 0.4048

Ties	P\D	2	3	4	5	6	7	8	9	10	J	Q	K	A
	2								11					
	3	lower than the initial dealer's card							12					
	4	~~6~~	~~7~~	~~8~~	~~9~~	~~10~~	11	12	13	14	14	14	14	15
	5								14					
	6								15					
	7								~~16~~					
	8								~~17~~					
	9								~~18~~					
	10		not reachable by player's second card {						~~19~~					
	J								~~19~~					
	Q								~~19~~					
	K								~~19~~					
	A								~~20~~					

Example 2: Player gets 10, Dealer gets 3. This example gives 3 possible sums: 12, 13, 14 for a tie-value

This way, in the table the point (Player's 1st. Card, Dealer's 1st Card) = (10, 3) will be P(12)+P(13)+P(14)= 0.2729

Ties	P\D	2	3	4	5	6	7	8	9	10	J	Q	K	A
	2		~~5~~											
	3		~~6~~											
	4		~~7~~											
	5		~~8~~		} lower than the initial player's card									
	6		~~9~~											
	7		~~10~~											
	8		~~11~~											
	9		12											
	10		13											
	J		13				not reachable by player's second card							
	Q	12	13	14	~~15~~	~~16~~	~~17~~	~~18~~	~~19~~	~~20~~	~~20~~	~~20~~	~~20~~	~~21~~
	K		13											
	A		14											

General table where each cell represents a probability of a tie for each pair of first cards Player x Dealer

This way, in the table the point P(Player's 1st. Card, Dealer's 1st Card) :

Ties	P\D	2	3	4	5	6	7	8	9	10-K	A
	2	0.3943	0.3898	0.3778	0.3613	0.3373	0.3088	0.2729	0.2324	0.1844	0.0960
	3	0.3898	0.4783	0.4663	0.4498	0.4258	0.3973	0.3613	0.3208	0.2729	0.1844
	4	0.3778	0.4663	0.5502	0.5337	0.5097	0.4813	0.4453	0.4048	0.3568	0.2684
	5	0.3613	0.4498	0.5337	0.6162	0.5922	0.5637	0.5277	0.4873	0.4393	0.3508
	6	0.3373	0.4258	0.5097	0.5922	0.6642	0.6357	0.5997	0.5592	0.5112	0.4228
	7	0.3088	0.3973	0.4813	0.5637	0.6357	0.7001	0.6642	0.6237	0.5757	0.4873
	8	0.2729	0.3613	0.4453	0.5277	0.5997	0.6642	0.7241	0.6837	0.6357	0.5472
	9	0.2324	0.3208	0.4048	0.4873	0.5592	0.6237	0.6837	0.7856	0.7376	0.6492
	10-K	0.1844	0.2729	0.3568	0.4393	0.5112	0.5757	0.6357	0.7376	0.7856	0.6972
	A	0.0960	0.1844	0.2684	0.3508	0.4228	0.4873	0.5472	0.6492	0.6972	0.7016

Based on this table, our strategy will always bet for TIEs as the gain is 10:1, except when the initial initial pair of card is (A,2) or (2, A), which will be diregarded for | player - dealer | = 9

Second DECISION is to double or to keep origial bet for winning.

For doubling decision, we will evaluate the odds of winning also based on the first cards dealt, as follows:

Fact: For the person with the face card, the possible maximun total hand is Card + 11

Example 1: Player gets 4, Dealer gets 9. This example gives 5 possible sums: 11, 12, 13, 14, 15 for a tie-value.

This way, in the table the point (Player's 1st. Card, Dealer's 1st Card) = (4, 9) will be P(MinCard+11+1) + ... P(LargerCard + 11), to Dealer

Ties	P\D	2	3	4	5	6	7	8	9	10	J	Q	K	A
	2								~~11~~					
	3	lower than the initial dealer's card							~~12~~					
	4	~~6~~	~~7~~	~~8~~	~~9~~	~~10~~	~~11~~	~~12~~	~~13~~	~~14~~	~~14~~	~~14~~	~~14~~	~~15~~
	5								~~14~~					
	6								~~15~~					
	7								16					
	8								17					
	9								18					
	10			advantage of dealer over player {					19					
	J								19					
	Q								19					
	K								19					
	A								20					

Example 2: Player gets 10, Dealer gets 3. This example gives 3 possible sums: 12, 13, 14 for a tie-value

This way, in the table the point (Player's 1st. Card, Dealer's 1st Card) = (4, 9) will be P(MinCard+11+1) + ... P(LargerCard + 11), to Player

Ties	P\D	2	3	4	5	6	7	8	9	10	J	Q	K	A
	2		~~5~~											
	3		~~6~~											
	4		~~7~~											
	5		~~8~~		} lower than the initial player's card									
	6		~~9~~											
	7		~~10~~											
	8		~~11~~											
	9		~~12~~											
	10		~~13~~											
	J		~~13~~				advantage of player over the dealer							
	Q	~~12~~	~~13~~	~~14~~	15	16	17	18	19	20	20	20	20	21
	K		~~13~~											
	A		~~14~~											

Adv.	P\D	2	3	4	5	6	7	8	9	10-K	A
	2	0.0000	-0.0885	-0.1724	-0.2549	-0.3268	-0.3913	-0.4513	-0.5532	-0.6012	-0.6057
	3	0.0885	0.0000	-0.0840	-0.1664	-0.2384	-0.3028	-0.3628	-0.4648	-0.5127	-0.5172
	4	0.1724	0.0840	0.0000	-0.0825	-0.1544	-0.2189	-0.2789	-0.3808	-0.4288	-0.4333
	5	0.2549	0.1664	0.0825	0.0000	-0.0720	-0.1364	-0.1964	-0.2984	-0.2744	-0.3508
	6	0.3268	0.2384	0.1544	0.0720	0.0000	-0.0645	-0.1244	-0.2264	-0.2744	-0.2789
	7	0.3913	0.3028	0.2189	0.1364	0.0645	0.0000	-0.0600	-0.1619	-0.2099	-0.2144
	8	0.4513	0.3628	0.2789	0.1964	0.1244	0.0600	0.0000	-0.1019	-0.1499	-0.1544
	9	0.5532	0.4648	0.3808	0.2984	0.2264	0.1619	0.1019	0.0000	-0.0480	-0.0525
	10-K	0.6012	0.5127	0.4288	0.3463	0.2744	0.2099	0.1499	0.0480	0.0000	-0.0045
	A	0.6057	0.5172	0.4333	0.3508	0.2789	0.2144	0.1544	0.0525	0.0045	0.0000

Legend:

Advantage to Player

Advantage to Dealer

No advantage

General strategy will be to double everytime the first cards are greater than the dealers' first card.

In the case of tie for first cards Player vs Dealer, we will assume no doubling will be made.

As we will always bet for Tie, as per our strategy, if cards are the same, we will ALSO Double when it's tie below:

Decision:	P\D	2	3	4	5	6	7	8	9	10-K	A
Double	2	TIE/DD	KP	KP	KP	KP	KP	KP	KP	KP	KP
or Not	3	DD	TIE/DD	KP	KP	KP	KP	KP	KP	KP	KP
	4	DD	DD	TIE/DD	KP	KP	KP	KP	KP	KP	KP
	5	DD	DD	DD	TIE/DD	KP	KP	KP	KP	KP	KP
	6	DD	DD	DD	DD	TIE/DD	KP	KP	KP	KP	KP
	7	DD	DD	DD	DD	DD	TIE/DD	KP	KP	KP	KP
	8	DD	DD	DD	DD	DD	DD	TIE/DD	KP	KP	KP
	9	DD	DD	DD	DD	DD	DD	DD	TIE/DD	KP	KP
	10-K	DD	DD	DD	DD	DD	DD	DD	DD	TIE/DD	KP
	A	DD	DD	DD	DD	DD	DD	DD	DD	DD	TIE/DD

Legend:

TIE meas equal changes

DD means double initial bet

KP means keep original bet

Strategy for whether or not going to war after a TIE (non-bet) will depend uppon there was a double-bet. DD => NO WAR

Let's evaluate possible outcomes from different strategies and simulate a large number of rounds

For each possible dollar outcome, an histogram will show the weight of each dollar outcome.

Initial bet will always be 2 (even number) $value chips for standard reference

See attached earning_paths.png

Functions: p(y), F(Y), E(Y), sd(Y), plots f(Y) and F(Y)

This game will do the reverse way: create a program to run this game 10 millions of times, and register the p(y) for each possible outcome.

Check attached the code to be used on data-generated report.

https://app.mindmup.com/map/_free/2021/11/56a2a1b0483811ec9708eb12a2b87af4

Data set as histogram per $2 initial bet.												
Number of times a possible result was actually obtained per row-sequence of million rounds												
Result: Y	**-6**	**-4**	**-2**	**2**	**4**	**6**	**8**	**16**	**18**		**# Games**	
	141401	317868	5926	77788	20	380140	5770	46794	24293		10^6	
	141369	317897	5958	78563	12	379532	5834	46253	24582		10^6	
	141427	317654	6009	78888	13	378858	5910	46560	24681		10^6	
	141512	317990	5926	78768	20	378975	5882	46248	24679		10^6	
	140840	317173	5924	78912	15	379892	5845	46707	24692		10^6	
	141958	316405	5949	78557	16	380274	5912	46307	24622		10^6	
	141520	317324	5829	78220	17	379877	5894	46667	24652		10^6	
	140995	317419	5851	78460	12	280579	5893	46305	24486		10^6	
	142133	317544	5867	78388	15	379433	5873	46086	24661		10^6	
	141936	317125	5990	78228	14	379512	5800	46542	24853		10^6	
Average:	**141509**	**317440**	**5923**	**78477**	**15**	**369707**	**5861**	**46447**	**24620**		**1000000**	
y	**-6**	**-4**	**-2**	**2**	**4**	**6**	**8**	**16**	**18**			
py(y)	**0.14151**	**0.31744**	**0.00592**	**0.07848**	**0.00002**	**0.36971**	**0.00586**	**0.04645**	**0.02462**		**1.00000**	
y * py(y)	-0.849054	-1.269759	-0.011845	0.1569544	0.0000616	2.2182432	0.0468904	0.7431504	0.4431618	(dot prod) =	**1.4778**	**E(Y)**
# of Paths												**Average**
total_win	463693	463926	463653	463623	464647	464741	463989	464931	463691	463540		**464043**
total_lose	465146	465169	465036	465373	463892	464255	464624	464209	465489	465003		**464820**
total_tie	71161	70905	71311	71004	71461	71004	71387	70860	70820	71457		**71137**
total_war	35	34	31	41	29	43	39	33	33	30		**35**
total_nowar	39	36	39	36	33	32	29	36	40	32		**35**
total_win2	24	12	13	20	15	16	17	12	15	10		**15**
total_lose2	10	19	15	19	12	25	20	20	15	16		**17**
total_tie2	1	3	3	2	2	2	2	1	3	4		**2**
myGrandBala	1536896	1531844	1536278	1529828	1550184	1540960	1543256	1541866	1526984	1539930		**1537803**
winPerGame	1.5369	1.5318	1.5363	1.5298	1.5502	1.5410	1.5433	1.5419	1.5270	1.5399	Actual	**1.5378**

NET WINNINGS:

Actual **Average Winnings per Game:** **1.5378**

Let's suppose that the casino knows that the actual net earning per $2 chip is $1.54, which is in the player's favor in order to bring the advantage back to them, it is charged a $8 fee so you can play with $10-value chips.

Y = 5X - 8

E(Y) = E(5*X - 8) = 5*E(X) - 8 = 5* 1.47 - 8 = 7.7 - 8 = -0.65

Var(Y) = Var(5X − 8) = 25 Var(X) , where Var(X) = E(X^2) - E(X)^2

sd(Y) = sqrt(Var(Y)

```
# Double-War Data Generator  # Note: time to run 10^6 rounds is about 1 min
# Code made available at:
https://github.com/hedravBands/2P81_Probability_with_R/tree/main/JackWar

# Global variables
max_tries   <- 10^6
total_win   <- 0
total_lose  <- 0
total_tie   <- 0
total_war   <- 0
total_nowar <- 0
total_win2  <- 0
total_lose2 <- 0
total_tie2  <- 0
MasterDeck  <- c(2:10, 10, 10, 10, 11, 2:10, 10, 10, 10, 11, 2:10, 10, 10, 10, 11,
2:10, 10, 10, 10, 11)
results     <- c(1:28)*0  #histogram using indexes as results: where 10 is even, 11 is
1x gain, 1 is -1x loss


# Balance:
myGrandBalance    <- 0
myOriginalBalance <- 0
myOriginalTieBet  <- 0
myOriginalBet     <- 2

# Strategy
onTieDifference   <- 8

################

# Util functions

newDeck <- function(){
  # face values: 2:10, J, K, Q, A  times  4 = 52 cards
  return(MasterDeck)
}

# Draw card at any time during a game
drawCard <- function(){
  idx <- sample(1:52, 1)
  # re-sample until card is valid (value not 0)
  while (WorkingDeck[idx] == 0) {
    idx <- sample(1:52, 1)
  }
  card <- WorkingDeck[idx]
  WorkingDeck <- replace(WorkingDeck, idx, 0)   #set card as drawn (value 0)

  return(card)
}

# Decision-Maker 1: Absolute difference less than or equal to onTieDifference
shouldBetOnTie <- function(h, d){
  return((abs(h[1]-d[1]) <= onTieDifference))
}

# Decision-Maker 2: Player's first card is greater than or equal to Dealer's
shouldDoubleBet <- function(h, d){
```

```
  return((h[1] - d[1]) >= 0)
}

# Decision-Maker 3: Decision whether or not go to WAR: DD => no war
shouldGoToWar <- function(dd){
  return(!dd)
}


# CHECK VICTORY FOR 1ST ROUND, 1 WIN, 0 TIE, -1 LOSE
isThisAVictory <- function(h, d){
  if ((h[1]+h[2]) - (d[1]+d[2])  > 0) {return(1)}
  if ((h[1]+h[2]) - (d[1]+d[2]) == 0) {return(0)}
  if ((h[1]+h[2]) - (d[1]+d[2]) < 0) {return(-1)}
}
###################################

## PLAY SEVERAL TIMES
for(i in 1:max_tries){

# since only 1 deck is available, every round needs a new deck
WorkingDeck <- newDeck()

# MONEY RELATED
myBalance   <- myOriginalBalance
myBet       <- myOriginalBet
myTieBet    <- myOriginalTieBet

# GET INITIAL CARDS
hand    <- c(drawCard(), 0)
dealer  <- c(drawCard(), 0)

# DECISION-MAKER 1: BET ON TIE?
isBetOnTie = shouldBetOnTie(hand, dealer)
if (isBetOnTie) {myTieBet <- myBet}

# DECISION-MAKER 2: DOUBLE ORIGINAL BET?
isDoubleBet <- shouldDoubleBet(hand, dealer)
if (isDoubleBet) {myBet <- 2*myBet}

# Play for ROUND 1
hand[2]   <- drawCard()
dealer[2] <- drawCard()

# DECISION-MAKER 3: GO TO WAR? Strategy: DD => NO WAR
isThisWar <- shouldGoToWar(isDoubleBet)

result <- isThisAVictory(hand, dealer)

#  EVALUATE EACH POSSIBLE RESULT: WIN, LOSE, TIE, NO_WAR, WAR: WIN2, LOSE2, TIE2
if (result == 1) {
  total_win <- total_win + 1
  myBalance <- myBet*2 - myTieBet
  myGrandBalance <- myGrandBalance + myBalance
  index <- 10 + myBalance
  results <- replace(results, index, results[index] + 1)
}
else if (result == -1){
```

```
  total_lose <- total_lose + 1
  myBalance <- result*(myBet + myTieBet)
  myGrandBalance <- myGrandBalance + myBalance
  index <- result*myBalance
  results <- replace(results, index, results[index] + 1)
}
else{
      total_tie <- total_tie + 1
      #cases for tie 1st. round
      if (isBetOnTie) {
        myBalance <- myTieBet*10 - myBet
        myGrandBalance <- myGrandBalance + myBalance
        index <- 10 + myBalance
        results <- replace(results, index, results[index] + 1)
      } #end win by tie result
      else{
            if (!isThisWar) {
            total_nowar <- total_nowar + 1
            myBalance <- as.integer((-1)*myBet/2) #will become index: integer
            myGrandBalance <- myGrandBalance + myBalance
            index <- (-1)*myBalance
            results <- replace(results, index, results[index] + 1)
            } #end not-war
            else {
                    total_war <- total_war + 1
                    myBet <- 2*myBet

                    # GET ANOTHER PAIR OF CARDS (AND DISCART THE OLD PAIRs)
                    hand2    <- c(drawCard(), drawCard())
                    dealer2  <- c(drawCard(), drawCard())

                    # GET NEW RESULT AND ANALYSE
                    result2 <- isThisAVictory(hand2, dealer2)

                    # ANALYSIS FOR THE SECOND ROUND, AFTER WAR: WIN2,LOSE2,TIE2
                    if (result2 == 1) {
                      total_win2 <- total_win2 + 1
                      myBalance <- myBet
                      myGrandBalance <- myGrandBalance + myBalance
                      index <- 10 + myBalance
                      results <- replace(results, index, results[index] + 1)

                    } else if (result2 == -1){
                      total_lose2 <- total_lose2 + 1
                      myBalance <- result2*(myBet)
                      myGrandBalance <- myGrandBalance + myBalance
                      index <- result2*myBalance
                      results <- replace(results, index, results[index] + 1)

                    } else  {
                      total_tie2 <- total_tie2 + 1
                      myBalance <- myBet*2
                      myGrandBalance <- myGrandBalance + myBalance
                      index <- 10 + myBalance
                      results <- replace(results, index, results[index] + 1)
                    }
            }
```

```
      } #end tie-cases
}
} # end for-loop

winPerGame = myGrandBalance/max_tries

##############

total_win
total_lose
total_tie
total_war
total_nowar
total_win2
total_lose2
total_tie2
myGrandBalance
winPerGame


results

################

#
# total_win
# [1] 463540
# > total_lose
# [1] 465003
# > total_tie
# [1] 71457
# > total_war
# [1] 30
# > total_nowar
# [1] 32
# > total_win2
# [1] 10
# > total_lose2
# [1] 16
# > total_tie2
# [1] 4
# > myGrandBalance
# [1] 1539930
# > winPerGame
# [1] 1.53993
# >
#   >
#   > results
# [1]      0   5990      0 317125      0 141936      0      0      0      0      0
78228      0     10      0 379512      0   5804      0      0      0
# [22]      0      0      0      0  46542      0  24853
# >
#   > ################
```