```
# Double-War Data Generator
# Note: time to run 10^6 rounds is about 1 min

# 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
# >
#   > ################
```