Download My Battleship Code Here:

https://drive.google.com/drive/folders/12anovhuBP-mO4727EtYyv1vFZR0-JZGD?usp=sharing

```
#' Play a Game of Battleship
#'
#' @description play_bs() takes an input of a length = 2 character vector that describes who is playing:
1 human, 1 ai, 2 humans, or 2 ais and simulates a game of Battleship. If both players are ais, the
player must input a battleship object with positioned fleets.
#'
#' @usage play_bs(players,battleship=NULL,strength1 = 0, strength2 = 0)
#'
#' @param players A length = 2 character vector, describes who is playing
#'     For example, c("human", "human"), c("ai","ai")
#' @param battleship A battleship object, needed anytime an ai is playing
#' @param strength1 A number, representing the strength of the first ai, 0 by default
#' @param strength2 A number, representing the strength of the second ai, 0 by default
#' @param ... Additional optional arguments
#'
#' @return A list containing the winning admiral's name, the number of turns the game took to complete,
the number of hits the losing admiral was rewarded, the number of unsunk ships of the winning admiral,
and a logical value determing if the winner's last boat was the length = 2 Patrol Boat, and a logical
value determining if the losing admiral's last boat was the length = 2 Patrol Boat
#' @export
#'
#' @seealso \link{battleship} to see the requirements for a battleship object
#'     \link{plot.battleship} and \link{plot.fleet} to see the battleship or fleet object plotted
#'     \link{summary.battleship} and \link{summary.fleet} to see summary statistics on a battleship or
fleet object
#'
#' @examples
#'play_bs(players = c("human", "human"))
play_bs <- function(players,battleship=NULL,strength1 = 0, strength2 = 0,...){
  if(length(battleship) > 0 && length(battleship$fleet[[1]]$ship[[1]][[3]]) < 1){
    stop("Ships must be positioned before playing!")
  }
  if(is.numeric(strength1) == FALSE | is.numeric(strength2) == FALSE | length(strength1) != 1 |
length(strength2) != 1){
    stop("Strength arguments must be a single numeric object!")
  }
  if(any(players == "Human")){ players[which(players == "Human")] <- "human"}
  if(any(players == "HUMAN")){ players[which(players == "HUMAN")] <- "human"}
```

```
  if(any(players == "AI")){ players[which(players == "AI")] <- "ai"}
  if(any(players == "Ai")){ players[which(players == "Ai")] <- "ai"}
  if(any(players != "human") && any(players != "ai")){
    stop("Check players vector, must be either 'human' or 'ai'!")
  }
  players <- sort(players,decreasing=T)
  names <- c("Player 1","Player 2") ; winner <- character() ; unsunk <- numeric(2) ; losthit <-
numeric() ; PB <- logical() ; lastPB <- logical()
  if(any(players=="human")){
    if(isFALSE(if(interactive()) askYesNo("Is default ocean and default ships okay?"))){
      battleship$fleet[[1]]$ocean[1] <- readline(prompt = "How many rows would you like for Player 1's
ocean?")
      battleship$fleet[[1]]$ocean[2] <- readline(prompt = "How many columns would you like for Player
1's ocean?")
      battleship$fleet[[2]]$ocean[1] <- readline(prompt = "How many rows would you like for Player 2's
ocean?")
      battleship$fleet[[2]]$ocean[2] <- readline(prompt = "How many columns would you like for Player
2's ocean?")
      nship1 <- readline(prompt = "How many ships would you like Player 1 to have?")
      for(i in 1:nship1){
        battleship$fleet[[1]]$ship[[i]][[1]] <- readline(prompt = paste("What name would you like for
Ship #",i,"?"))
        battleship$fleet[[1]]$ship[[i]][[2]] <- as.numeric(readline(prompt = paste("What size would you
like for Ship #",i,"?")))
      }
      nship2 <- readline(prompt = "How many ships would you like Player 2 to have?")
      for(i in 1:nship2){
        battleship$fleet[[2]]$ship[[i]][[1]] <- readline(prompt = paste("What name would you like for
Ship #",i,"?"))
        battleship$fleet[[2]]$ship[[i]][[2]] <- as.numeric(readline(prompt = paste("What size would you
like for Ship #",i,"?")))
      }
    }
  }#end setting ocean and ships
  if(players[1]=="human"){
    positions1 <- vector("list",length(battleship$fleet[[1]]$ship))
    plot(0:max(battleship$fleet[[1]]$ocean),0:max(battleship$fleet[[1]]
$ocean),type="n",xaxt="n",yaxt="n",xlab="",ylab="")
    title("Player 1's Battleship")
```

```
    abline(h=0:battleship$fleet[[1]]$ocean[1],v=0:battleship$fleet[[1]]$ocean[2])
    axis(2,at=0.5:(battleship$fleet[[1]]$ocean[1]-0.5),labels=letters[battleship$fleet[[1]]$ocean[1]:
1],tick=F)
    axis(3,at=0.5:(battleship$fleet[[2]]$ocean[1]-0.5),labels=1:battleship$fleet[[1]]
$ocean[2],line=-1,tick=F)
    cat("Player 1, please place your ships",fill=T)
    for(i in 1:length(battleship$fleet[[1]]$ship)){
      cat("Ship Name:",battleship$fleet[[1]]$ship[[i]][[1]],fill=T)
      cat("Ship Size:",battleship$fleet[[1]]$ship[[i]][[2]],fill=T)
      positions1[[i]][1] <- readline(prompt = "Where would you like the front of your ship to be?")
      positions1[[i]][2] <- readline(prompt = "Where would you like the end of your ship to be?")
      plot.fleet(battleship$fleet[[1]],positions=positions1[1:i])
    }
    try(while(isFALSE(if(interactive()) askYesNo("Is this placement ok?"))){
      positions1 <- vector("list",length(battleship$fleet[[1]]$ship))
      plot(0:max(battleship$fleet[[1]]$ocean),0:max(battleship$fleet[[1]]
$ocean[1]),type="n",xaxt="n",yaxt="n",xlab="",ylab="")
      title("Player 1's Battleship")
      abline(h=0:battleship$fleet[[1]]$ocean[1],v=0:battleship$fleet[[1]]$ocean[2])
      axis(2,at=0.5:(battleship$fleet[[1]]$ocean[1]-0.5),labels=letters[battleship$fleet[[1]]$ocean[1]:
1],tick=F)
      axis(3,at=0.5:(battleship$fleet[[2]]$ocean[1]-0.5),labels=1:battleship$fleet[[1]]
$ocean[2],line=-1,tick=F)
      cat("Player 1, please replace your ships",fill=T)
      for(i in 1:length(battleship$fleet[[1]]$ship)){
        cat("Ship Name:",battleship$fleet[[1]]$ship[[i]][[1]],fill=T)
        cat("Ship Size:",battleship$fleet[[1]]$ship[[i]][[2]],fill=T)
        positions1[[i]][1] <- readline(prompt = "Where would you like the front of your ship to be?")
        positions1[[i]][2] <- readline(prompt = "Where would you like the end of your ship to be?")
        plot(battleship$fleet[[1]],positions1[1:i])
      }
    })
    hits1 <- position_fleet(battleship$fleet[[1]],positions1)
    cat("Thank you, Player 1",fill=T)
  }else if(players[1]!="human"){
    hits1 <- position_fleet(battleship$fleet[[1]])
  }
  if(players[2]=="human"){
    cat("Please give your opponent the computer so they can place their ships",fill=T)
```

```
    positions2 <- vector("list",length(battleship$fleet[[2]]$ship))
    plot(0:max(battleship$fleet[[2]]$ocean),0:max(battleship$fleet[[2]]
$ocean),type="n",xaxt="n",yaxt="n",xlab="",ylab="")
    title("Player 2's Battleship")
    abline(h=0:battleship$fleet[[2]]$ocean[1],v=0:battleship$fleet[[2]]$ocean[2])
    axis(2,at=0.5:(battleship$fleet[[2]]$ocean[1]-0.5),labels=letters[battleship$fleet[[2]]$ocean[1]:
1],tick=F)
    axis(3,at=0.5:(battleship$fleet[[2]]$ocean[1]-0.5),labels=1:battleship$fleet[[2]]
$ocean[2],line=-1,tick=F)
    cat("Player 2, please place your ships",fill=T)
    for(i in 1:length(battleship$fleet[[2]]$ship)){
      cat("Ship Name:",battleship$fleet[[2]]$ship[[i]][[1]],fill=T)
      cat("Ship Size:",battleship$fleet[[2]]$ship[[i]][[2]],fill=T)
      positions2[[i]][1] <- readline(prompt = "Where would you like the front of your ship to be?")
      positions2[[i]][2] <- readline(prompt = "Where would you like the end of your ship to be?")
      plot(battleship$fleet[[2]],positions2[1:i])
    }
    try(while(isFALSE(if(interactive()) askYesNo("Is this placement ok?"))){
      positions2 <- vector("list",length(battleship$fleet[[1]]$ship))
      plot(0:max(battleship$fleet[[2]]$ocean),0:max(battleship$fleet[[2]]
$ocean[1]),type="n",xaxt="n",yaxt="n",xlab="",ylab="")
      title("Player 2's Battleship")
      abline(h=0:battleship$fleet[[2]]$ocean[1],v=0:battleship$fleet[[2]]$ocean[2])
      axis(2,at=0.5:(battleship$fleet[[2]]$ocean[1]-0.5),labels=letters[battleship$fleet[[2]]$ocean[1]:
1],tick=F)
      axis(3,at=0.5:(battleship$fleet[[2]]$ocean[1]-0.5),labels=1:battleship$fleet[[2]]
$ocean[2],line=-1,tick=F)
      cat("Player 2, please replace your ships",fill=T)
      for(i in 1:length(battleship$fleet[[2]]$ship)){
        cat("Ship Name:",battleship$fleet[[2]]$ship[[i]][[1]],fill=T)
        cat("Ship Size:",battleship$fleet[[2]]$ship[[i]][[2]],fill=T)
        positions2[[i]][1] <- readline(prompt = "Where would you like the front of your ship to be?")
        positions2[[i]][2] <- readline(prompt = "Where would you like the end of your ship to be?")
        plot(battleship$fleet[[2]],positions2[1:i])
      }
    })
    hits2 <- position_fleet(battleship$fleet[[2]],positions2)
    cat("Thank you, Player 2",fill=T)
  }else if(players[2]!="human"){
```

```
    hits2 <- position_fleet(battleship$fleet[[2]])
  }
  ta1 <- rep(letters[1:battleship$fleet[[1]]$ocean[1]],battleship$fleet[[1]]$ocean[2])
  tb1 <- numeric() ; y1 <- 1 ; z1 <- battleship$fleet[[1]]$ocean[2]
  for(i in 1:battleship$fleet[[1]]$ocean[2]){
    tb1[y1:z1] <- rep(i,battleship$fleet[[1]]$ocean[1])
    y1 <- y1+battleship$fleet[[1]]$ocean[1]
    z1 <- z1+battleship$fleet[[1]]$ocean[1]
  }
  allspots1 <- paste(ta1,"-",tb1)
  ta2 <- rep(letters[1:battleship$fleet[[2]]$ocean[1]],battleship$fleet[[2]]$ocean[1])
  tb2 <- numeric() ; y2 <- 1 ; z2 <- battleship$fleet[[2]]$ocean[2]
  for(i in 1:z2){
    tb2[y2:z2] <- rep(i,battleship$fleet[[2]]$ocean[1])
    y2 <- y2 + battleship$fleet[[2]]$ocean[1]
    z2 <- z2 + battleship$fleet[[2]]$ocean[1]
  }
  allspots2 <- paste(ta2,"-",tb2)
  hmp1 <- tibble(x=rep(battleship$fleet[[1]]$ocean[[1]]:1,battleship$fleet[[1]]
$ocean[[2]]),y=rep(1:battleship$fleet[[1]]$ocean[[2]],rep(battleship$fleet[[1]]
$ocean[[1]],battleship$fleet[[1]]$ocean[[2]])),target=allspots1,hit=rep("miss",battleship$fleet[[1]]
$ocean[[1]]*battleship$fleet[[1]]$ocean[[2]]),shipnum=rep(0,battleship$fleet[[1]]
$ocean[[1]]*battleship$fleet[[1]]$ocean[[2]]))
  hmp2 <- tibble(x=rep(battleship$fleet[[2]]$ocean[[1]]:1,battleship$fleet[[2]]
$ocean[[2]]),y=rep(1:battleship$fleet[[2]]$ocean[[2]],rep(battleship$fleet[[2]]
$ocean[[1]],battleship$fleet[[2]]$ocean[[2]])),target=allspots2,hit=rep("miss",battleship$fleet[[2]]
$ocean[[1]]*battleship$fleet[[2]]$ocean[[2]]),shipnum=rep(0,battleship$fleet[[2]]
$ocean[[1]]*battleship$fleet[[2]]$ocean[[2]]))
  #gameplay
  nums1 <- length(battleship$fleet[[1]]$ship) * 2
  for(i in 1:length(battleship$fleet[[1]]$ship)){
    name <- paste("s",i,sep = "")
    extra <- numeric()
    for(j in 1:(battleship$fleet[[1]]$ship[[i]][[2]] - 2)){
      if(battleship$fleet[[1]]$ship[[i]][[2]] == 2){break}
      nums1 <- nums1 + 1
      extra <- c(extra, nums1)
    }
    assign(name,c( ((2*i) - 1) , (2*i),extra) )
```

```
  }
  nums2 <- length(battleship$fleet[[2]]$ship) * 2
  for(i in 1:length(battleship$fleet[[2]]$ship)){
    name <- paste("s",i,2,sep = "")
    extra <- numeric()
    for(j in 1:(battleship$fleet[[2]]$ship[[i]][[2]] - 2)){
      if(battleship$fleet[[2]]$ship[[i]][[2]] == 2){break}
      nums2 <- nums2 + 1
      extra <- c(extra, nums2)
    }
    assign(name,c( ((2*i) - 1) , (2*i),extra) )
  }
  if(all(players!="human")){ #2ai
    strength1 <- strength1 ; strength2 <- strength2
    pos1 <- list() ; pos2 <- list()
    for(i in 1:length(hits1$ship)){
      pos1[2*i-1] <- hits1$ship[[i]]$position[1]
      pos1[2*i] <- hits1$ship[[i]]$position[2]
    }
    for(i in 1:length(hits2$ship)){
      pos2[2*i-1] <- hits2$ship[[i]]$position[1]
      pos2[2*i] <- hits2$ship[[i]]$position[2]
    }
    pos1num <- vector("list",length(pos1))
    pos2num <- vector("list",length(pos2))
    for(i in 1:length(pos1)){
      pos1num[[i]][[2]] <- battleship$fleet[[1]]$ocean[[1]]-(which(letters==tolower(substr(pos1[[i]]
[1],start=1,stop=1)))-1)
      pos1num[[i]][[1]] <- as.numeric(substr(pos1[[i]][1],start=5,stop=6))
    }
    for(i in 1:length(pos2)){
      pos2num[[i]][[2]] <- battleship$fleet[[2]]$ocean[[1]]-(which(letters==tolower(substr(pos2[[i]]
[1],start=1,stop=1)))-1)
      pos2num[[i]][[1]] <- as.numeric(substr(pos2[[i]][1],start=5,stop=6))
    }
    l1 <- length(pos1num)
    for(i in 1:(length(pos1num)/2)){
      for(j in 1:(pos1num[[2*i-1]][[2]]-pos1num[[2*i]][[2]]-1)){
```

```
        if(max(pos1num[[2*i-1]][[2]],pos1num[[2*i]][[2]])-min(pos1num[[2*i-1]][[2]],pos1num[[2*i]]
[[2]])<2){break}
        a <- pos1num[[2*i]][[1]]
        b <- pos1num[[2*i-1]][[2]]-j
        l1 <- l1 + 1
        pos1num[[l1]] <- c(a,b)
      }
    }
    l2 <- length(pos2num)
    for(i in 1:(length(pos2num)/2)){
      for(j in 1:(pos2num[[2*i-1]][[2]]-pos2num[[2*i]][[2]]-1)){
        if(pos2num[[2*i-1]][[2]]-(pos2num[[2*i]][[2]])<2){break}
        a <- pos2num[[2*i]][[1]]
        b <- pos2num[[2*i-1]][[2]]-j
        l2 <- l2 + 1
        pos2num[[l2]] <- c(a,b)
      }
    }#return every position where there should be a hit
    for(i in 1:length(pos1num)){
      hmp1 <- hmp1 %>% mutate(hit = replace(hit, which(y==pos1num[[i]][[1]] & x==pos1num[[i]][2]),
"hit"))
    }
    for(i in 1:length(s1)){hmp1 <- hmp1 %>% mutate(shipnum = replace(shipnum, which(y==pos1num[[ s1[i]
]][1] & x==pos1num[[ s1[i] ]][2]), 1))}
    for(i in 1:length(s2)){hmp1 <- hmp1 %>% mutate(shipnum = replace(shipnum, which(y==pos1num[[ s2[i]
]][1] & x==pos1num[[ s2[i] ]][2]), 2))}
    for(i in 1:length(s3)){hmp1 <- hmp1 %>% mutate(shipnum = replace(shipnum, which(y==pos1num[[ s3[i]
]][1] & x==pos1num[[ s3[i] ]][2]), 3))}
    for(i in 1:length(s4)){hmp1 <- hmp1 %>% mutate(shipnum = replace(shipnum, which(y==pos1num[[ s4[i]
]][1] & x==pos1num[[ s4[i] ]][2]), 4))}
    for(i in 1:length(s5)){hmp1 <- hmp1 %>% mutate(shipnum = replace(shipnum, which(y==pos1num[[ s5[i]
]][1] & x==pos1num[[ s5[i] ]][2]), 5))}
    for(i in 1:length(pos2num)){
      hmp2 <- hmp2 %>% mutate(hit = replace(hit, which(y==pos2num[[i]][[1]] & x==pos2num[[i]][2]),
"hit"))
    }#return a tibble with the numbers, letters, and hit or miss
    for(i in 1:length(s12)){a2 <- s12[i] ; hmp2 <- hmp2 %>% mutate(shipnum = replace(shipnum,
which(y==pos2num[[a2]][1] & x==pos2num[[a2]][2]), 1))}
```

```
    for(i in 1:length(s22)){a2 <- s22[i] ; hmp2 <- hmp2 %>% mutate(shipnum = replace(shipnum,
which(y==pos2num[[a2]][1] & x==pos2num[[a2]][2]), 2))}
    for(i in 1:length(s32)){a2 <- s32[i] ; hmp2 <- hmp2 %>% mutate(shipnum = replace(shipnum,
which(y==pos2num[[a2]][1] & x==pos2num[[a2]][2]), 3))}
    for(i in 1:length(s42)){a2 <- s42[i] ; hmp2 <- hmp2 %>% mutate(shipnum = replace(shipnum,
which(y==pos2num[[a2]][1] & x==pos2num[[a2]][2]), 4))}
    for(i in 1:length(s52)){a2 <- s52[i] ; hmp2 <- hmp2 %>% mutate(shipnum = replace(shipnum,
which(y==pos2num[[a2]][1] & x==pos2num[[a2]][2]), 5))}
    p1 <- -1 ; p2 <- 0 ; turns <- numeric(2) ; p1moves <-
ai_904795402(battleship,strength1,pn=1,spitall=T) ; p2moves <-
ai_904795402(battleship,strength2,pn=2,spitall=T) ; p2hn <- character()
    m1 <- 1 ; m2 <- 1 ; p1hits <- numeric(1) ; p2hits <- numeric(1) ; strength1 <- strength1 ; strength2
<- strength2
    ss1 <- NULL ; ss2 <- NULL
    while(length(winner)!=1){
      p1 <- p1 + 2
      turns[1] <- turns[1] + 1
      battleship$history[p1,1] <- "Player 1"
      battleship$history[p1,2] <- "Player 2"
      battleship$history[p1,3] <- p1moves[m1]
      battleship$history[p1,4] <- filter(hmp2,target==p1moves[m1])$hit
      if(filter(hmp2,target==p1moves[m1])$hit=="hit"){
        p1hits <- p1hits + 1
        a <- 1
        s <- filter(hmp2,target==p1moves[m1])$shipnum
        ss1 <- c(ss1,s)
        while(battleship$fleet[[2]]$ship[[s]]$hits[a]==TRUE){
          a <- a + 1
          if(all(battleship$fleet[[2]]$ship[[s]]$hits==TRUE)){break}
        }
        battleship$fleet[[2]]$ship[[s]]$hits[a] <- TRUE
        if(all(battleship$fleet[[2]]$ship[[s]]$hits==TRUE)){
          battleship$fleet[[2]]$ship[[s]]$sunk <- TRUE
        }
      }
      m1 <- m1 + 1
      if(p1hits==length(pos2num)){
        winner <- "Player 1"
      }
```

```
  if(length(winner)>0){break}
  p2 <- p2 + 2
  turns[2] <- turns[2] + 1
  battleship$history[p2,1] <- "Player 2"
  battleship$history[p2,2] <- "Player 1"
  battleship$history[p2,3] <- p2moves[m2]
  battleship$history[p2,4] <- filter(hmp1,target==p2moves[m2])$hit
  if(filter(hmp1,target==p2moves[m2])$hit=="hit"){
    p2hits <- p2hits + 1
    a <- 1
    s <- filter(hmp1,target==p2moves[m2])$shipnum
    ss2 <- c(ss2,s)
    while(battleship$fleet[[1]]$ship[[s]]$hits[a]==TRUE){
      a <- a + 1
      if(all(battleship$fleet[[1]]$ship[[s]]$hits==TRUE)){break}
    }
    battleship$fleet[[1]]$ship[[s]]$hits[a] <- TRUE
    if(all(battleship$fleet[[1]]$ship[[s]]$hits==TRUE)){
      battleship$fleet[[1]]$ship[[s]]$sunk <- TRUE
    }
  }
  m2 <- m2 + 1
  if(p2hits==length(pos1num)){
    winner <- "Player 2"
  }
  if(length(winner)>0){break}
}
for(i in 1:length(battleship$fleet[[1]]$ship)){
  if(battleship$fleet[[1]]$ship[[i]]$sunk==F){
    unsunk[1] <- unsunk[1] + 1
  }
}
for(i in 1:length(battleship$fleet[[2]]$ship)){
  if(battleship$fleet[[2]]$ship[[i]]$sunk==F){
    unsunk[2] <- unsunk[2] + 1
  }
}
loserhits <- min(p1hits,p2hits)
turns <- max(turns)
```

```
    unsunk <- max(unsunk)
    PB <- battleship$fleet[[which(names==winner)]]$ship[[5]]$sunk
    d <- which(names!=winner)
    sss <- c(ss1,ss2)
    ssss <- last(sss[d])
    if(ssss == 2){
      lastPB <- TRUE
    }else{
      lastPB <- FALSE
    }
  }else if(all(players=="human")){ #2 human
    pos1 <- list() ; pos2 <- list()
    for(i in 1:length(hits1$ship)){
      pos1[2*i-1] <- hits1$ship[[i]]$position[1]
      pos1[2*i] <- hits1$ship[[i]]$position[2]
    }
    for(i in 1:length(hits2$ship)){
      pos2[2*i-1] <- hits2$ship[[i]]$position[1]
      pos2[2*i] <- hits2$ship[[i]]$position[2]
    }
    pos1num <- vector("list",length(pos1))
    pos2num <- vector("list",length(pos2))
    for(i in 1:length(pos1)){
      pos1num[[i]][[2]] <- battleship$fleet[[1]]$ocean[[1]]-(which(letters==tolower(substr(pos1[[i]]
[1],start=1,stop=1)))-1)
      pos1num[[i]][[1]] <- as.numeric(substr(pos1[[i]][1],start=5,stop=6))
    }
    for(i in 1:length(pos2)){
      pos2num[[i]][[2]] <- battleship$fleet[[2]]$ocean[[1]]-(which(letters==tolower(substr(pos2[[i]]
[1],start=1,stop=1)))-1)
      pos2num[[i]][[1]] <- as.numeric(substr(pos2[[i]][1],start=5,stop=6))
    }
    l1 <- length(pos1num)
    for(i in 1:(length(pos1num)/2)){
      if(pos1num[[2*i]][1]==pos1num[[2*i -1]][1]){#vertical ship
        for(j in 1:((max(pos1num[[2*i-1]][[2]],pos1num[[2*i]][[2]])-min(pos1num[[2*i-1]]
[[2]],pos1num[[2*i]][[2]]))-1)){
          if(max(pos1num[[2*i-1]][[2]],pos1num[[2*i]][[2]])-min(pos1num[[2*i-1]][[2]],pos1num[[2*i]]
[[2]])<2){break}
```

```
          a <- pos1num[[2*i]][[1]]
          b <- max(pos1num[[2*i-1]][[2]],pos1num[[2*i]][[2]])-j
          l1 <- l1 + 1
          pos1num[[l1]] <- c(a,b)
        }
      }else if(pos1num[[2*i]][2]==pos1num[[2*i -1]][2]){ #horizontal ship
        for(j in 1:((max(pos1num[[2*i]][[1]],pos1num[[2*i-1]][[1]])-min(pos1num[[2*i]]
[[1]],pos1num[[2*i-1]][[1]]))-1)){
          if(max(pos1num[[2*i]][[1]],pos1num[[2*i-1]][[1]])-min(pos1num[[2*i]][[1]],pos1num[[2*i-1]]
[[1]])<2){break}
          b <- pos1num[[2*i-1]][[2]]
          a <- max(pos1num[[2*i]][[1]],pos1num[[2*i-1]][[1]])-j
          l1 <- l1 + 1
          pos1num[[l1]] <- c(a,b)
        }
      }
    }
    l2 <- length(pos2num)
    for(i in 1:(length(pos2num)/2)){
      if(pos2num[[2*i]][1]==pos2num[[2*i -1]][1]){#vertical ship
        for(j in 1:((max(pos2num[[2*i-1]][[2]],pos2num[[2*i]][[2]])-min(pos2num[[2*i-1]]
[[2]],pos2num[[2*i]][[2]]))-1)){
          if(max(pos2num[[2*i-1]][[2]],pos2num[[2*i]][[2]])-min(pos2num[[2*i-1]][[2]],pos2num[[2*i]]
[[2]])<2){break}
          a <- pos2num[[2*i]][[1]]
          b <- max(pos2num[[2*i-1]][[2]],pos2num[[2*i]][[2]])-j
          l2 <- l2 + 1
          pos2num[[l2]] <- c(a,b)
        }
      }else if(pos2num[[2*i]][2]==pos2num[[2*i -1]][2]){ #horizontal ship
        for(j in 1:((max(pos2num[[2*i]][[1]],pos2num[[2*i-1]][[1]])-min(pos2num[[2*i]]
[[1]],pos2num[[2*i-1]][[1]]))-1)){
          if(max(pos2num[[2*i]][[1]],pos2num[[2*i-1]][[1]])-min(pos2num[[2*i]][[1]],pos2num[[2*i-1]]
[[1]])<2){break}
          b <- pos2num[[2*i-1]][[2]]
          a <- max(pos2num[[2*i]][[1]],pos2num[[2*i-1]][[1]])-j
          l2 <- l2 + 1
          pos2num[[l2]] <- c(a,b)
        }
```

```
      }
    }
    for(i in 1:length(pos1num)){
      hmp1 <- hmp1 %>% mutate(hit = replace(hit, which(y==pos1num[[i]][[1]] & x==pos1num[[i]][2]),
"hit"))
    }
    for(i in 1:length(s1)){a1 <- s1[i] ; hmp1 <- hmp1 %>% mutate(shipnum = replace(shipnum,
which(y==pos1num[[a1]][1] & x==pos1num[[a1]][2]), 1))}
    for(i in 1:length(s2)){a1 <- s2[i] ; hmp1 <- hmp1 %>% mutate(shipnum = replace(shipnum,
which(y==pos1num[[a1]][1] & x==pos1num[[a1]][2]), 2))}
    for(i in 1:length(s3)){a1 <- s3[i] ; hmp1 <- hmp1 %>% mutate(shipnum = replace(shipnum,
which(y==pos1num[[a1]][1] & x==pos1num[[a1]][2]), 3))}
    for(i in 1:length(s4)){a1 <- s4[i] ; hmp1 <- hmp1 %>% mutate(shipnum = replace(shipnum,
which(y==pos1num[[a1]][1] & x==pos1num[[a1]][2]), 4))}
    for(i in 1:length(s5)){a1 <- s5[i] ; hmp1 <- hmp1 %>% mutate(shipnum = replace(shipnum,
which(y==pos1num[[a1]][1] & x==pos1num[[a1]][2]), 5))}
    for(i in 1:length(pos2num)){
      hmp2 <- hmp2 %>% mutate(hit = replace(hit, which(y==pos2num[[i]][[1]] & x==pos2num[[i]][2]),
"hit"))
    }#return a tibble with the numbers, letters, and hit or miss
    for(i in 1:length(s1)){a2 <- s1[i] ; hmp2 <- hmp2 %>% mutate(shipnum = replace(shipnum,
which(y==pos2num[[a2]][1] & x==pos2num[[a2]][2]), 1))}
    for(i in 1:length(s2)){a2 <- s2[i] ; hmp2 <- hmp2 %>% mutate(shipnum = replace(shipnum,
which(y==pos2num[[a2]][1] & x==pos2num[[a2]][2]), 2))}
    for(i in 1:length(s3)){a2 <- s3[i] ; hmp2 <- hmp2 %>% mutate(shipnum = replace(shipnum,
which(y==pos2num[[a2]][1] & x==pos2num[[a2]][2]), 3))}
    for(i in 1:length(s4)){a2 <- s4[i] ; hmp2 <- hmp2 %>% mutate(shipnum = replace(shipnum,
which(y==pos2num[[a2]][1] & x==pos2num[[a2]][2]), 4))}
    for(i in 1:length(s5)){a2 <- s5[i] ; hmp2 <- hmp2 %>% mutate(shipnum = replace(shipnum,
which(y==pos2num[[a2]][1] & x==pos2num[[a2]][2]), 5))}
    p1 <- -1 ; p2 <- 0 ; turns <- numeric(2) ; p1moves <- character() ; p2moves <- character() ; p1hn <-
character() ; p2hn <- character()
    m1 <- 1 ; m2 <- 1 ; p1hits <- numeric(1) ; p2hits <- numeric(1)
    plot(0:max(battleship$fleet[[2]]$ocean),0:max(battleship$fleet[[2]]
$ocean),type="n",xaxt="n",yaxt="n",xlab="",ylab="")
    title("Player 2's Ocean")
    abline(h=0:battleship$fleet[[2]]$ocean[1],v=0:battleship$fleet[[2]]$ocean[2])
    axis(2,at=0.5:(battleship$fleet[[2]]$ocean[1]-0.5),labels=letters[battleship$fleet[[2]]$ocean[1]:
1],tick=F)
```

```
    axis(3,at=0.5:(battleship$fleet[[2]]$ocean[1]-0.5),labels=1:battleship$fleet[[2]]
$ocean[2],line=-1,tick=F)
    player1history <- recordPlot()
    ss1 <- NULL ; ss2 <- NULL
    plot(0:max(battleship$fleet[[1]]$ocean),0:max(battleship$fleet[[1]]
$ocean),type="n",xaxt="n",yaxt="n",xlab="",ylab="")
    title("Player 1's Ocean")
    abline(h=0:battleship$fleet[[1]]$ocean[1],v=0:battleship$fleet[[1]]$ocean[2])
    axis(2,at=0.5:(battleship$fleet[[1]]$ocean[1]-0.5),labels=letters[battleship$fleet[[1]]$ocean[1]:
1],tick=F)
    axis(3,at=0.5:(battleship$fleet[[1]]$ocean[1]-0.5),labels=1:battleship$fleet[[1]]
$ocean[2],line=-1,tick=F)
    player2history <- recordPlot()
    while(length(winner)!=1){
      p1 <- p1 + 2
      turns[1] <- turns[1] + 1
      print(player1history)
      p1moves[m1] <- readline(prompt = "Player 1, what is your move?")
      if(any(unlist(strsplit(p1moves[m1],split=NULL))=="-")){
        if(length(unlist(strsplit(p1moves[m1],split=NULL)))<=5){
          p1moves[m1] <- paste(tolower(unlist(strsplit(p1moves[m1],split=NULL))
[1]),"-",unlist(strsplit(p1moves[m1],split=NULL))[5])
        }
      }
      if(any(strsplit(p1moves[m1],split=NULL)!="-")){
        if(length(unlist(strsplit(p1moves[m1],split=NULL)))==2){
          p1moves[m1] <- paste(tolower(unlist(strsplit(p1moves[m1],split=NULL))
[1]),"-",unlist(strsplit(p1moves[m1],split=NULL))[2])
        }else if(length(unlist(strsplit(tries[i],split=NULL)))==3){
          p1moves[m1] <- paste(tolower(unlist(strsplit(p1moves[m1],split=NULL))
[1]),"-",unlist(strsplit(p1moves[m1],split=NULL))[2:3])
        }
      }
      battleship$history[p1,1] <- "Player 1"
      battleship$history[p1,2] <- "Player 2"
      battleship$history[p1,3] <- p1moves[m1]
      battleship$history[p1,4] <- filter(hmp2,target==p1moves[m1])$hit
      if(filter(hmp2,target==p1moves[m1])$hit=="miss"){
        player1history
```

```
        points(filter(hmp2,target==p1moves[m1])$y - 0.5,filter(hmp2,target==p1moves[m1])$x -
0.5,col="black")
        player1history <- recordPlot()
      }
      if(filter(hmp2,target==p1moves[m1])$hit=="hit"){
        p1hits <- p1hits + 1
        a <- 1
        s <- filter(hmp2,target==p1moves[m1])$shipnum
        ss1 <- c(ss1,s)
        player1history
        points(filter(hmp2,target==p1moves[m1])$y - 0.5,filter(hmp2,target==p1moves[m1])$x -
0.5,col="red")
        player1history <- recordPlot()
        while(battleship$fleet[[2]]$ship[[s]]$hits[a]==TRUE){
          a <- a + 1
          if(all(battleship$fleet[[2]]$ship[[s]]$hits==TRUE)){break}
        }
        battleship$fleet[[2]]$ship[[s]]$hits[a] <- TRUE
        if(all(battleship$fleet[[2]]$ship[[s]]$hits==TRUE)){
          battleship$fleet[[2]]$ship[[s]]$sunk <- TRUE
        }
      }
      player1history
      cat(paste("Your Shot:",battleship$history[p1,3]),fill=T)
      cat(paste("Hit/Miss:",battleship$history[p1,4]),fill=T)
      print(player1history)
      m1 <- m1 + 1
      if(p1hits==length(pos2num)){
        winner <- "Player 1"
      }
      if(length(winner)>0){break}
      p2 <- p2 + 2
      turns[2] <- turns[2] + 1
      print(player2history)
      p2moves[m2] <- readline(prompt = "Player 2, what is your move?")
      if(any(unlist(strsplit(p2moves[m2],split=NULL))=="-")){
        if(length(unlist(strsplit(p2moves[m2],split=NULL)))<=5){
          p2moves[m2] <- paste(tolower(unlist(strsplit(p2moves[m2],split=NULL))
[1]),"-",unlist(strsplit(p2moves[m2],split=NULL))[5])
```

```
        }
      }
      if(any(strsplit(p2moves[m2],split=NULL)!="-")){
        if(length(unlist(strsplit(p2moves[m2],split=NULL)))==2){
          p2moves[m2] <- paste(tolower(unlist(strsplit(p2moves[m2],split=NULL))
[1]),"-",unlist(strsplit(p2moves[m2],split=NULL))[2])
        }else if(length(unlist(strsplit(tries[i],split=NULL)))==3){
          p2moves[m2] <- paste(tolower(unlist(strsplit(p2moves[m2],split=NULL))
[1]),"-",unlist(strsplit(p2moves[m2],split=NULL))[2:3])
        }
      }
      battleship$history[p2,1] <- "Player 2"
      battleship$history[p2,2] <- "Player 1"
      battleship$history[p2,3] <- p2moves[m2]
      battleship$history[p2,4] <- filter(hmp1,target==p2moves[m2])$hit
      player2history
      points(filter(hmp1,target==p2moves[m2])$y - 0.5,filter(hmp1,target==p2moves[m2])$x -
0.5,col="black")
      player2history <- recordPlot()
      print(player2history)
      if(filter(hmp1,target==p2moves[m2])$hit=="hit"){
        p2hits <- p2hits + 1
        a <- 1
        s <- filter(hmp1,target==p2moves[m2])$shipnum
        ss2 <- c(ss2,s)
        player2history
        points(filter(hmp1,target==p2moves[m2])$y - 0.5,filter(hmp1,target==p2moves[m2])$x -
0.5,col="red")
        player2history <- recordPlot()
        print(player2history)
        while(battleship$fleet[[1]]$ship[[s]]$hits[a]==TRUE){
          a <- a + 1
          if(all(battleship$fleet[[1]]$ship[[s]]$hits==TRUE)){break}
        }
        battleship$fleet[[1]]$ship[[s]]$hits[a] <- TRUE
        if(all(battleship$fleet[[1]]$ship[[s]]$hits==TRUE)){
          battleship$fleet[[1]]$ship[[s]]$sunk <- TRUE
        }
      }
```

```
    player2history
    cat(paste("Your Shot:",battleship$history[p2,3]),fill=TRUE)
    cat(paste("Hit/Miss:",battleship$history[p2,4]),fill=TRUE)
    print(player1history)
    m2 <- m2 + 1
    if(p2hits==length(pos1num)){
      winner <- "Player 2"
    }
    if(length(winner)>0){break}
  }
  for(i in 1:length(battleship$fleet[[1]]$ship)){
    if(battleship$fleet[[1]]$ship[[i]]$sunk==F){
      unsunk[1] <- unsunk[1] + 1
    }
  }
  for(i in 1:length(battleship$fleet[[2]]$ship)){
    if(battleship$fleet[[2]]$ship[[i]]$sunk==F){
      unsunk[2] <- unsunk[2] + 1
    }
  }
  loserhits <- min(p1hits,p2hits)
  turns <- max(turns)
  unsunk <- max(unsunk)
  PB <- battleship$fleet[[which(names==winner)]]$ship[[5]]$sunk
  d <- which(names!=winner)
  sss <- c(ss1,ss2)
  ssss <- last(sss[d])
  if(ssss == 2){
    lastPB <- TRUE
  }else{
    lastPB <- FALSE
  }
  cat("Thanks for playing!",fill=T)
}else if(players[1]=="human"){ #one human, one ai
  pos1 <- list() ; pos2 <- list()
  for(i in 1:length(hits1$ship)){
    pos1[2*i-1] <- hits1$ship[[i]]$position[1]
    pos1[2*i] <- hits1$ship[[i]]$position[2]
  }
```

```
    for(i in 1:length(hits2$ship)){
      pos2[2*i-1] <- hits2$ship[[i]]$position[1]
      pos2[2*i] <- hits2$ship[[i]]$position[2]
    }
    pos1num <- vector("list",length(pos1))
    pos2num <- vector("list",length(pos2))
    for(i in 1:length(pos1)){
      pos1num[[i]][[2]] <- battleship$fleet[[1]]$ocean[[1]]-(which(letters==tolower(substr(pos1[[i]]
[1],start=1,stop=1)))-1)
      pos1num[[i]][[1]] <- as.numeric(substr(pos1[[i]][1],start=5,stop=6))
    }
    for(i in 1:length(pos2)){
      pos2num[[i]][[2]] <- battleship$fleet[[2]]$ocean[[1]]-(which(letters==tolower(substr(pos2[[i]]
[1],start=1,stop=1)))-1)
      pos2num[[i]][[1]] <- as.numeric(substr(pos2[[i]][1],start=5,stop=6))
    }
    l1 <- length(pos1num)
    for(i in 1:(length(pos1num)/2)){
      if(pos1num[[2*i]][1]==pos1num[[2*i -1]][1]){#vertical ship
        for(j in 1:(pos1num[[2*i-1]][[2]]-pos1num[[2*i]][[2]]-1)){
          if(pos1num[[2*i-1]][[2]]-(pos1num[[2*i]][[2]])<2){break}
          a <- pos1num[[2*i]][[1]]
          b <- pos1num[[2*i-1]][[2]]-j
          l1 <- l1 + 1
          pos1num[[l1]] <- c(a,b)
        }
      }else if(pos1num[[2*i]][2]==pos1num[[2*i -1]][2]){ #horizontal ship
        for(j in 1:(pos1num[[2*i]][[1]]-pos1num[[2*i-1]][[1]]-1)){
          if(pos1num[[2*i]][[1]]-(pos1num[[2*i-1]][[1]])<2){break}
          b <- pos1num[[2*i-1]][[2]]
          a <- pos1num[[2*i]][[1]]-j
          l1 <- l1 + 1
          pos1num[[l1]] <- c(a,b)
        }
      }
    }
    l2 <- length(pos2num)
    for(i in 1:(length(pos2num)/2)){
      if(pos2num[[2*i]][1]==pos2num[[2*i -1]][1]){#vertical ship
```

```
        for(j in 1:(pos2num[[2*i-1]][[2]]-pos2num[[2*i]][[2]]-1)){
          if(pos2num[[2*i-1]][[2]]-(pos2num[[2*i]][[2]])<2){break}
          a <- pos2num[[2*i]][[1]]
          b <- pos2num[[2*i-1]][[2]]-j
          l2 <- l2 + 1
          pos2num[[l2]] <- c(a,b)
        }
      }else if(pos2num[[2*i]][2]==pos2num[[2*i -1]][2]){ #horizontal ship
        for(j in 1:(pos2num[[2*i]][[1]]-pos2num[[2*i-1]][[1]]-1)){
          if(pos2num[[2*i]][[1]]-(pos1num[[2*i-1]][[1]])<2){break}
          b <- pos2num[[2*i-1]][[2]]
          a <- pos2num[[2*i]][[1]]-j
          l2 <- l2 + 1
          pos2num[[l2]] <- c(a,b)
        }
      }
    } #return every position where there should be a hit
    for(i in 1:length(pos1num)){
      hmp1 <- hmp1 %>% mutate(hit = replace(hit, which(y==pos1num[[i]][[1]] & x==pos1num[[i]][2]),
"hit"))
    }
    for(i in 1:length(s1)){a1 <- s1[i] ; hmp1 <- hmp1 %>% mutate(shipnum = replace(shipnum,
which(y==pos1num[[a1]][1] & x==pos1num[[a1]][2]), 1))}
    for(i in 1:length(s2)){a1 <- s2[i] ; hmp1 <- hmp1 %>% mutate(shipnum = replace(shipnum,
which(y==pos1num[[a1]][1] & x==pos1num[[a1]][2]), 2))}
    for(i in 1:length(s3)){a1 <- s3[i] ; hmp1 <- hmp1 %>% mutate(shipnum = replace(shipnum,
which(y==pos1num[[a1]][1] & x==pos1num[[a1]][2]), 3))}
    for(i in 1:length(s4)){a1 <- s4[i] ; hmp1 <- hmp1 %>% mutate(shipnum = replace(shipnum,
which(y==pos1num[[a1]][1] & x==pos1num[[a1]][2]), 4))}
    for(i in 1:length(s5)){a1 <- s5[i] ; hmp1 <- hmp1 %>% mutate(shipnum = replace(shipnum,
which(y==pos1num[[a1]][1] & x==pos1num[[a1]][2]), 5))}
    for(i in 1:length(pos2num)){
      hmp2 <- hmp2 %>% mutate(hit = replace(hit, which(y==pos2num[[i]][[1]] & x==pos2num[[i]][2]),
"hit"))
    }#return a tibble with the numbers, letters, and hit or miss
    for(i in 1:length(s1)){a2 <- s1[i] ; hmp2 <- hmp2 %>% mutate(shipnum = replace(shipnum,
which(y==pos2num[[a2]][1] & x==pos2num[[a2]][2]), 1))}
    for(i in 1:length(s2)){a2 <- s2[i] ; hmp2 <- hmp2 %>% mutate(shipnum = replace(shipnum,
which(y==pos2num[[a2]][1] & x==pos2num[[a2]][2]), 2))}
```

```
    for(i in 1:length(s3)){a2 <- s3[i] ; hmp2 <- hmp2 %>% mutate(shipnum = replace(shipnum,
which(y==pos2num[[a2]][1] & x==pos2num[[a2]][2]), 3))}
    for(i in 1:length(s4)){a2 <- s4[i] ; hmp2 <- hmp2 %>% mutate(shipnum = replace(shipnum,
which(y==pos2num[[a2]][1] & x==pos2num[[a2]][2]), 4))}
    for(i in 1:length(s5)){a2 <- s5[i] ; hmp2 <- hmp2 %>% mutate(shipnum = replace(shipnum,
which(y==pos2num[[a2]][1] & x==pos2num[[a2]][2]), 5))}
    p1 <- -1 ; p2 <- 0 ; turns <- numeric(2) ; p1moves <- character() ; p2moves <-
ai_904795402(battleship,strength2,pn=2,spitall=T) ; p1hn <- character() ; p2hn <- character()
    m1 <- 1 ; m2 <- 1 ; p1hits <- numeric(1) ; p2hits <- numeric(1)
    ss1 <- NULL ; ss2 <- NULL
    while(length(winner)!=1){
      p1 <- p1 + 2
      turns[1] <- turns[1] + 1
      p1moves[m1] <- readline(prompt = "Player 1, what is your move?")
      if(any(unlist(strsplit(p1moves[m1],split=NULL))=="-")){
        if(length(unlist(strsplit(p1moves[m1],split=NULL)))<=5){
          p1moves[m1] <- paste(tolower(unlist(strsplit(p1moves[m1],split=NULL))
[1]),"-",unlist(strsplit(p1moves[m1],split=NULL))[5])
        }
      }
      if(any(strsplit(p1moves[m1],split=NULL)!="-")){
        if(length(unlist(strsplit(p1moves[m1],split=NULL)))==2){
          p1moves[m1] <- paste(tolower(unlist(strsplit(p1moves[m1],split=NULL))
[1]),"-",unlist(strsplit(p1moves[m1],split=NULL))[2])
        }else if(length(unlist(strsplit(tries[i],split=NULL)))==3){
          p1moves[m1] <- paste(tolower(unlist(strsplit(p1moves[m1],split=NULL))
[1]),"-",unlist(strsplit(p1moves[m1],split=NULL))[3])
        }
      }
      battleship$history[p1,1] <- "Player 1"
      battleship$history[p1,2] <- "Player 2"
      battleship$history[p1,3] <- p1moves[m1]
      battleship$history[p1,4] <- filter(hmp2,target==p1moves[m1])$hit
      if(filter(hmp2,target==p1moves[m1])$hit=="hit"){
        p1hits <- p1hits + 1
        a <- 1
        s <- filter(hmp2,target==p1moves[m1])$shipnum
        ss1 <- c(ss1,s)
        while(battleship$fleet[[2]]$ship[[s]]$hits[a]==TRUE){
```

```
    a <- a + 1
    if(all(battleship$fleet[[2]]$ship[[s]]$hits==TRUE)){break}
  }
  battleship$fleet[[2]]$ship[[s]]$hits[a] <- TRUE
  if(all(battleship$fleet[[2]]$ship[[s]]$hits==TRUE)){
    battleship$fleet[[2]]$ship[[s]]$sunk <- TRUE
  }
}
print(battleship$history[p1,])
m1 <- m1 + 1
if(p1hits==length(pos2num)){
  winner <- "Player 1"
}
if(length(winner)>0){break}
p2 <- p2 + 2
turns[2] <- turns[2] + 1
battleship$history[p2,1] <- "Player 2"
battleship$history[p2,2] <- "Player 1"
battleship$history[p2,3] <- p2moves[m2]
battleship$history[p2,4] <- filter(hmp1,target==p2moves[m2])$hit
if(filter(hmp1,target==p2moves[m2])$hit=="hit"){
  p2hits <- p2hits + 1
  a <- 1
  s <- filter(hmp1,target==p2moves[m2])$shipnum
  ss2 <- c(ss2,s)
  while(battleship$fleet[[1]]$ship[[s]]$hits[a]==TRUE){
    a <- a + 1
    if(all(battleship$fleet[[1]]$ship[[s]]$hits==TRUE)){break}
  }
  battleship$fleet[[1]]$ship[[s]]$hits[a] <- TRUE
  if(all(battleship$fleet[[1]]$ship[[s]]$hits==TRUE)){
    battleship$fleet[[1]]$ship[[s]]$sunk <- TRUE
  }
}
cat("This was Player 2's move",fill=T)
print(battleship$history[p2,])
m2 <- m2 + 1
if(p2hits==length(pos1num)){
  winner <- "Player 2"
```

```
      }
      if(length(winner)>0){break}
    }
    for(i in 1:length(battleship$fleet[[1]]$ship)){
      if(battleship$fleet[[1]]$ship[[i]]$sunk==F){
        unsunk[1] <- unsunk[1] + 1
      }
    }
    for(i in 1:length(battleship$fleet[[2]]$ship)){
      if(battleship$fleet[[2]]$ship[[i]]$sunk==F){
        unsunk[2] <- unsunk[2] + 1
      }
    }
    loserhits <- min(p1hits,p2hits)
    turns <- max(turns)
    unsunk <- max(unsunk)
    PB <- battleship$fleet[[which(names==winner)]]$ship[[5]]$sunk
    d <- which(names!=winner)
    sss <- c(ss1,ss2)
    ssss <- last(sss[d])
    if(ssss == 2){
      lastPB <- TRUE
    }else{
      lastPB <- FALSE
    }
    cat("Thanks for playing!")
  }
  list(winner=winner,turns=turns,loserhits=loserhits,unsunk=unsunk,winnerPB=PB,loserlastPB=lastPB)
}
```