

# chess

Geovert John D. Labita

2023-07-22

```
#loading
#chess board
board <- matrix("", nrow = 8, ncol = 8)
board[2, ] <- c("P_1", "P_2", "P_3", "P_4", "P_5", "P_6", "P_7", "P_8")
board[7, ] <- c("p_1", "p_2", "p_3", "p_4", "p_5", "p_6", "p_7", "p_8")
board[1, ] <- c("R_1", "H_1", "B_1", "Q", "K", "B_2", "H_2", "R_2")
board[8, ] <- c("r_1", "h_1", "b_1", "q", "k", "b_2", "h_2", "r_2")
rownames(board) <- 8:1
colnames(board) <- LETTERS[1:8]
black<-c("R_1", "H_1", "B_1", "Q", "K", "B_2", "H_2", "R_2", "P_1", "P_2", "P_3", "P_4", "P_5", "P_6", "P_7", "P_8")
white<-c("p_1", "p_2", "p_3", "p_4", "p_5", "p_6", "p_7", "p_8", "r_1", "h_1", "b_1", "q", "k", "b_2", "h_2", "r_2")
more_black<-c("R_1", "B_1", "Q", "B_2", "R_2")
more_white<-c("r_1", "b_1", "q", "b_2", "r_2")
b_score<-c(5,3,3,9,NA,3,3,5,1,1,1,1,1,1,1,1)
w_score<-c(1,1,1,1,1,1,1,5,3,3,9,NA,3,3,5)

#chess squares
files <- LETTERS[1:8]
ranks <- 8:1
chess_labels <- outer(ranks, files, FUN = function(r, f) paste0(f, r))
board2 <- t(matrix(chess_labels, nrow = 8, byrow = TRUE))

#move
move_piece <- function(board, from, to) {
  col_index <- function(x) match(substr(x, 1, 1), LETTERS[1:8])
  row_index <- function(x) 9 - as.numeric(substr(x, 2, 2))
  from_row <- row_index(from)
  from_col <- col_index(from)
  to_row <- row_index(to)
  to_col <- col_index(to)
  board[to_row, to_col] <- board[from_row, from_col]
  board[from_row, from_col] <- ""
  return(board)}

#king
king<-function(board,b_king,color1,color2){
bking_move1<-numeric()
bking_move2<-numeric()
bking_move3<-numeric()
bking_move4<-numeric()
bking_move5<-numeric()
bking_move6<-numeric()
```

```

bkings_move7<-numeric()
bkings_move8<-numeric()
if ((b_king[1]+1)<9 && (b_king[2]-1)>0){
  bkings_move1<-board[b_king[1]+1,b_king[2]-1]
  if (bkings_move1=="") {bkings_move1<-board2[b_king[1]+1,b_king[2]-1]}
if ((b_king[1]-1)>0 && (b_king[2]-1)>0){
  bkings_move2<-board[b_king[1]-1,b_king[2]-1]
  if (bkings_move2=="") {bkings_move2<-board2[b_king[1]-1,b_king[2]-1]}
if ((b_king[1]-1)>0 && (b_king[2]+1)<9){
  bkings_move3<-board[b_king[1]-1,b_king[2]+1]
  if (bkings_move3=="") {bkings_move3<-board2[b_king[1]-1,b_king[2]+1]}
if ((b_king[1]+1)<9 && (b_king[2]+1)<9){
  bkings_move4<-board[b_king[1]+1,b_king[2]+1]
  if (bkings_move4=="") {bkings_move4<-board2[b_king[1]+1,b_king[2]+1]}
if ((b_king[2]-1)>0){
  bkings_move5<-board[b_king[1],b_king[2]-1]
  if (bkings_move5=="") {bkings_move5<-board2[b_king[1],b_king[2]-1]}
if ((b_king[2]+1)<9){
  bkings_move6<-board[b_king[1],b_king[2]+1]
  if (bkings_move6=="") {bkings_move6<-board2[b_king[1],b_king[2]+1]}
if ((b_king[1]-1)>0{
  bkings_move7<-board[b_king[1]-1,b_king[2]]
  if (bkings_move7=="") {bkings_move7<-board2[b_king[1]-1,b_king[2]]}
if ((b_king[1]+1)<9){
  bkings_move8<-board[b_king[1]+1,b_king[2]]
  if (bkings_move8=="") {bkings_move8<-board2[b_king[1]+1,b_king[2]]}
king_movement<-c(bkings_move1,bkings_move2,bkings_move3,bkings_move4,bkings_move5,bkings_move6,bkings_move7,bk
if (all(color1==color2)==FALSE){
  l=length(king_movement)
  for (i in 1:l){
    if (king_movement[i] %in% color1 == TRUE){
      position<-match(king_movement[i],king_movement)
      king_movement[position]=NA}
    king_movement<-king_movement[!is.na(king_movement)]}
  return(king_movement)}

#queen
queen<-function(board,p_queen,color1,color2){
moves1<-numeric()
moves2<-numeric()
moves3<-numeric()
moves4<-numeric()
moves5<-numeric()
moves6<-numeric()
moves7<-numeric()
moves8<-numeric()
for (i in 1:7){
  if ((p_queen[1]+i)<9 && (p_queen[2]-i)>0){
    diagonal1<-board[p_queen[1]+i,p_queen[2]-i]
    if (diagonal1=="") {diagonal1=board2[p_queen[1]+i,p_queen[2]-i]}
    moves1<-append(moves1, diagonal1)}
  if ((p_queen[1]-i)>0 && (p_queen[2]-i)>0{
    diagonal2<-board[p_queen[1]-i,p_queen[2]-i]

```

```

    if (diagonal2=="") {diagonal2=board2[p_queen[1]-i,p_queen[2]-i]}
    moves2<-append(moves2, diagonal2)}
    if ((p_queen[1]-i)>0 && (p_queen[2]+i)<9){
        diagonal3<-board[p_queen[1]-i,p_queen[2]+i]
        if (diagonal3=="") {diagonal3=board2[p_queen[1]-i,p_queen[2]+i]}
        moves3<-append(moves3, diagonal3)}
    if ((p_queen[1]+i)<9 && (p_queen[2]+i)<9){
        diagonal4<-board[p_queen[1]+i,p_queen[2]+i]
        if (diagonal4=="") {diagonal4=board2[p_queen[1]+i,p_queen[2]+i]}
        moves4<-append(moves4, diagonal4)}
    if ((p_queen[2]-i)>0){
        horizontal1<-board[p_queen[1],p_queen[2]-i]
        if (horizontal1=="") {horizontal1=board2[p_queen[1],p_queen[2]-i]}
        moves5<-append(moves5, horizontal1)}
    if ((p_queen[2]+i)<9){
        horizontal2<-board[p_queen[1],p_queen[2]+i]
        if (horizontal2=="") {horizontal2=board2[p_queen[1],p_queen[2]+i]}
        moves6<-append(moves6, horizontal2)}
    if ((p_queen[1]-i)>0){
        vertical1<-board[p_queen[1]-i,p_queen[2]]
        if (vertical1=="") {vertical1=board2[p_queen[1]-i,p_queen[2]]}
        moves7<-append(moves7, vertical1)}
    if ((p_queen[1]+i)<9){
        vertical2<-board[p_queen[1]+i,p_queen[2]]
        if (vertical2=="") {vertical2=board2[p_queen[1]+i,p_queen[2]]}
        moves8<-append(moves8, vertical2)}}

f_moves1<-numeric()
f_moves2<-numeric()
f_moves3<-numeric()
f_moves4<-numeric()
f_moves5<-numeric()
f_moves6<-numeric()
f_moves7<-numeric()
f_moves8<-numeric()
for (i in 1:length(moves1)){
    if (length(moves1)==0){break}
    if (all(color1==color2) == TRUE){
        if (moves1[i] %in% white | moves1[i] %in% black){
            f_moves1[i]=moves1[i]
            break}}
    if (moves1[i] %in% color1 == TRUE){break}
    if (moves1[i] %in% color2 == TRUE){
        f_moves1[i]=moves1[i]
        break}
    f_moves1[i]=moves1[i]}
for (i in 1:length(moves2)){
    if (length(moves2)==0){break}
    if (all(color1==color2) == TRUE){
        if (moves2[i] %in% white | moves2[i] %in% black){
            f_moves2[i]=moves2[i]
            break}}
    if (moves2[i] %in% color1 == TRUE){break}
    if (moves2[i] %in% color2 == TRUE){

```

```

f_moves2[i]=moves2[i]
break}
f_moves2[i]=moves2[i]}
for (i in 1:length(moves3)){
if (length(moves3)==0){break}
if (all(color1==color2) == TRUE){
  if (moves3[i] %in% white | moves3[i] %in% black){
    f_moves3[i]=moves3[i]
    break}}
  if (moves3[i] %in% color1 == TRUE){break}
  if (moves3[i] %in% color2 == TRUE){
    f_moves3[i]=moves3[i]
    break}
  f_moves3[i]=moves3[i]}
for (i in 1:length(moves4)){
if (length(moves4)==0){break}
if (all(color1==color2) == TRUE){
  if (moves4[i] %in% white | moves4[i] %in% black){
    f_moves4[i]=moves4[i]
    break}}
  if (moves4[i] %in% color1 == TRUE){break}
  if (moves4[i] %in% color2 == TRUE){
    f_moves4[i]=moves4[i]
    break}
  f_moves4[i]=moves4[i]}
for (i in 1:length(moves5)){
if (length(moves5)==0){break}
if (all(color1==color2) == TRUE){
  if (moves5[i] %in% white | moves5[i] %in% black){
    f_moves5[i]=moves5[i]
    break}}
  if (moves5[i] %in% color1 == TRUE){break}
  if (moves5[i] %in% color2 == TRUE){
    f_moves5[i]=moves5[i]
    break}
  f_moves5[i]=moves5[i]}
for (i in 1:length(moves6)){
if (length(moves6)==0){break}
if (all(color1==color2) == TRUE){
  if (moves6[i] %in% white | moves6[i] %in% black){
    f_moves6[i]=moves6[i]
    break}}
  if (moves6[i] %in% color1 == TRUE){break}
  if (moves6[i] %in% color2 == TRUE){
    f_moves6[i]=moves6[i]
    break}
  f_moves6[i]=moves6[i]}
for (i in 1:length(moves7)){
if (length(moves7)==0){break}
if (all(color1==color2) == TRUE){
  if (moves7[i] %in% white | moves7[i] %in% black){
    f_moves7[i]=moves7[i]
    break}}}

```

```

if (moves7[i] %in% color1 == TRUE){break}
if (moves7[i] %in% color2 == TRUE){
  f_moves7[i]=moves7[i]
  break}
f_moves7[i]=moves7[i]}
for (i in 1:length(moves8)){
  if (length(moves8)==0){break}
  if (all(color1==color2) == TRUE){
    if (moves8[i] %in% white | moves8[i] %in% black){
      f_moves8[i]=moves8[i]
      break}}
  if (moves8[i] %in% color1 == TRUE){break}
  if (moves8[i] %in% color2 == TRUE){
    f_moves8[i]=moves8[i]
    break}
  f_moves8[i]=moves8[i]}
return(c(f_moves1,f_moves2,f_moves3,f_moves4,f_moves5,f_moves6,f_moves7,f_moves8))}

#rook
rook<-function(board,p_rook1,color1,color2){
moves9<-numeric()
moves10<-numeric()
moves11<-numeric()
moves12<-numeric()
for (i in 1:7){
  if ((p_rook1[2]-i)>0){
    horizontal1<-board[p_rook1[1],p_rook1[2]-i]
    if (horizontal1==""){horizontal1=board2[p_rook1[1],p_rook1[2]-i]}
    moves9<-append(moves9, horizontal1)}
  if ((p_rook1[2]+i)<9){
    horizontal2<-board[p_rook1[1],p_rook1[2]+i]
    if (horizontal2==""){horizontal2=board2[p_rook1[1],p_rook1[2]+i]}
    moves10<-append(moves10, horizontal2)}
  if ((p_rook1[1]-i)>0){
    vertical1<-board[p_rook1[1]-i,p_rook1[2]]
    if (vertical1==""){vertical1=board2[p_rook1[1]-i,p_rook1[2]]}
    moves11<-append(moves11, vertical1)}
  if ((p_rook1[1]+i)<9){
    vertical2<-board[p_rook1[1]+i,p_rook1[2]]
    if (vertical2==""){vertical2=board2[p_rook1[1]+i,p_rook1[2]]}
    moves12<-append(moves12, vertical2)}}
f_moves9<-numeric()
f_moves10<-numeric()
f_moves11<-numeric()
f_moves12<-numeric()
for (i in 1:length(moves9)){
  if (length(moves9)==0){break}
  if (all(color1==color2) == TRUE){
    if (moves9[i] %in% white | moves9[i] %in% black){
      f_moves9[i]=moves9[i]
      break}}
  if (moves9[i] %in% color1 == TRUE){break}
  if (moves9[i] %in% color2 == TRUE){

```

```

f_moves9[i]=moves9[i]
break}
f_moves9[i]=moves9[i]}
for (i in 1:length(moves10)){
if (length(moves10)==0){break}
if (all(color1==color2) == TRUE){
  if (moves10[i] %in% white | moves10[i] %in% black){
    f_moves10[i]=moves10[i]
    break}}
  if (moves10[i] %in% color1 == TRUE){break}
  if (moves10[i] %in% color2 == TRUE){
    f_moves10[i]=moves10[i]
    break}
f_moves10[i]=moves10[i]}
for (i in 1:length(moves11)){
if (length(moves11)==0){break}
if (all(color1==color2) == TRUE){
  if (moves11[i] %in% white | moves11[i] %in% black){
    f_moves11[i]=moves11[i]
    break}}
  if (moves11[i] %in% color1 == TRUE){break}
  if (moves11[i] %in% color2 == TRUE){
    f_moves11[i]=moves11[i]
    break}
f_moves11[i]=moves11[i]}
for (i in 1:length(moves12)){
if (length(moves12)==0){break}
if (all(color1==color2) == TRUE){
  if (moves12[i] %in% white | moves12[i] %in% black){
    f_moves12[i]=moves12[i]
    break}}
  if (moves12[i] %in% color1 == TRUE){break}
  if (moves12[i] %in% color2 == TRUE){
    f_moves12[i]=moves12[i]
    break}
f_moves12[i]=moves12[i]}
return(c(f_moves9,f_moves10,f_moves11,f_moves12))

#bishop
bishop<-function(board,p_bishop1,color1,color2){
moves17<-numeric()
moves18<-numeric()
moves19<-numeric()
moves20<-numeric()
for (i in 1:7){
  if ((p_bishop1[1]+i)<9 && (p_bishop1[2]-i)>0){
    diagonal1<-board[p_bishop1[1]+i,p_bishop1[2]-i]
    if (diagonal1==""){diagonal1=board2[p_bishop1[1]+i,p_bishop1[2]-i]}
    moves17<-append(moves17, diagonal1)}
  if ((p_bishop1[1]-i)>0 && (p_bishop1[2]-i)>0){
    diagonal2<-board[p_bishop1[1]-i,p_bishop1[2]-i]
    if (diagonal2==""){diagonal2=board2[p_bishop1[1]-i,p_bishop1[2]-i]}
    moves18<-append(moves18, diagonal2)}}

```

```

if ((p_bishop1[1]-i)>0 && (p_bishop1[2]+i)<9){
    diagonal3<-board[p_bishop1[1]-i,p_bishop1[2]+i]
    if (diagonal3==""){diagonal3=board2[p_bishop1[1]-i,p_bishop1[2]+i]}
    moves19<-append(moves19, diagonal3)}
if ((p_bishop1[1]+i)<9 && (p_bishop1[2]+i)<9){
    diagonal4<-board[p_bishop1[1]+i,p_bishop1[2]+i]
    if (diagonal4==""){diagonal4=board2[p_bishop1[1]+i,p_bishop1[2]+i]}
    moves20<-append(moves20, diagonal4)}}
```

f\_moves17<-numeric()  
f\_moves18<-numeric()  
f\_moves19<-numeric()  
f\_moves20<-numeric()

```

for (i in 1:length(moves17)){
    if (length(moves17)==0){break}
    if (all(color1==color2) == TRUE){
        if (moves17[i] %in% white | moves17[i] %in% black){
            f_moves17[i]=moves17[i]
            break}}
        if (moves17[i] %in% color1 == TRUE){break}
        if (moves17[i] %in% color2 == TRUE){
            f_moves17[i]=moves17[i]
            break}
        f_moves17[i]=moves17[i]}
for (i in 1:length(moves18)){
    if (length(moves18)==0){break}
    if (all(color1==color2) == TRUE){
        if (moves18[i] %in% white | moves18[i] %in% black){
            f_moves18[i]=moves18[i]
            break}}
        if (moves18[i] %in% color1 == TRUE){break}
        if (moves18[i] %in% color2 == TRUE){
            f_moves18[i]=moves18[i]
            break}
        f_moves18[i]=moves18[i]}
for (i in 1:length(moves19)){
    if (length(moves19)==0){break}
    if (all(color1==color2) == TRUE){
        if (moves19[i] %in% white | moves19[i] %in% black){
            f_moves19[i]=moves19[i]
            break}}
        if (moves19[i] %in% color1 == TRUE){break}
        if (moves19[i] %in% color2 == TRUE){
            f_moves19[i]=moves19[i]
            break}
        f_moves19[i]=moves19[i]}
for (i in 1:length(moves20)){
    if (length(moves20)==0){break}
    if (all(color1==color2) == TRUE){
        if (moves20[i] %in% white | moves20[i] %in% black){
            f_moves20[i]=moves20[i]
            break}}
        if (moves20[i] %in% color1 == TRUE){break}
        if (moves20[i] %in% color2 == TRUE){
```

```

    f_moves20[i]=moves20[i]
    break}
f_moves20[i]=moves20[i]}
return(c(f_moves17,f_moves18,f_moves19,f_moves20))}

#knight
horse<-function(board,p_horse1,color1,color2){
horse_move1a<-"sample"
horse_move1b<-"sample"
horse_move1c<-"sample"
horse_move1d<-"sample"
horse_move1e<-"sample"
horse_move1f<-"sample"
horse_move1g<-"sample"
horse_move1h<-"sample"
if ((p_horse1[2]+2)>4 && (p_horse1[2]-2)<5 && (p_horse1[1]+2)>4 && (p_horse1[1]-2)<5){
  horse_move1a<-board[p_horse1[1]-1,p_horse1[2]-2]
  horse_move1b<-board[p_horse1[1]-2,p_horse1[2]-1]
  horse_move1c<-board[p_horse1[1]-2,p_horse1[2]+1]
  horse_move1d<-board[p_horse1[1]-1,p_horse1[2]+2]
  horse_move1e<-board[p_horse1[1]+1,p_horse1[2]+2]
  horse_move1f<-board[p_horse1[1]+2,p_horse1[2]+1]
  horse_move1g<-board[p_horse1[1]+2,p_horse1[2]-1]
  horse_move1h<-board[p_horse1[1]+1,p_horse1[2]-2]
  if (horse_move1a==""){horse_move1a<-board2[p_horse1[1]-1,p_horse1[2]-2]}
  if (horse_move1b==""){horse_move1b<-board2[p_horse1[1]-2,p_horse1[2]-1]}
  if (horse_move1c==""){horse_move1c<-board2[p_horse1[1]-2,p_horse1[2]+1]}
  if (horse_move1d==""){horse_move1d<-board2[p_horse1[1]-1,p_horse1[2]+2]}
  if (horse_move1e==""){horse_move1e<-board2[p_horse1[1]+1,p_horse1[2]+2]}
  if (horse_move1f==""){horse_move1f<-board2[p_horse1[1]+2,p_horse1[2]+1]}
  if (horse_move1g==""){horse_move1g<-board2[p_horse1[1]+2,p_horse1[2]-1]}
  if (horse_move1h==""){horse_move1h<-board2[p_horse1[1]+1,p_horse1[2]-2]}}
if (p_horse1[1]==2 && p_horse1[2]>2 && p_horse1[2]<7){
  horse_move1a<-board[p_horse1[1]-1,p_horse1[2]-2]
  horse_move1d<-board[p_horse1[1]-1,p_horse1[2]+2]
  horse_move1e<-board[p_horse1[1]+1,p_horse1[2]+2]
  horse_move1f<-board[p_horse1[1]+2,p_horse1[2]+1]
  horse_move1g<-board[p_horse1[1]+2,p_horse1[2]-1]
  horse_move1h<-board[p_horse1[1]+1,p_horse1[2]-2]
  if (horse_move1a==""){horse_move1a<-board2[p_horse1[1]-1,p_horse1[2]-2]}
  if (horse_move1d==""){horse_move1d<-board2[p_horse1[1]-1,p_horse1[2]+2]}
  if (horse_move1e==""){horse_move1e<-board2[p_horse1[1]+1,p_horse1[2]+2]}
  if (horse_move1f==""){horse_move1f<-board2[p_horse1[1]+2,p_horse1[2]+1]}
  if (horse_move1g==""){horse_move1g<-board2[p_horse1[1]+2,p_horse1[2]-1]}
  if (horse_move1h==""){horse_move1h<-board2[p_horse1[1]+1,p_horse1[2]-2]}}
if (p_horse1[1]==1 && p_horse1[2]>2 && p_horse1[2]<7){
  horse_move1e<-board[p_horse1[1]+1,p_horse1[2]+2]
  horse_move1f<-board[p_horse1[1]+2,p_horse1[2]+1]
  horse_move1g<-board[p_horse1[1]+2,p_horse1[2]-1]
  horse_move1h<-board[p_horse1[1]+1,p_horse1[2]-2]
  if (horse_move1e==""){horse_move1e<-board2[p_horse1[1]+1,p_horse1[2]+2]}
  if (horse_move1f==""){horse_move1f<-board2[p_horse1[1]+2,p_horse1[2]+1]}
  if (horse_move1g==""){horse_move1g<-board2[p_horse1[1]+2,p_horse1[2]-1]}}

```



```

if (horse_move1b=="") {horse_move1b<-board2[p_horse1[1]-2,p_horse1[2]-1]}
if (horse_move1c=="") {horse_move1c<-board2[p_horse1[1]-2,p_horse1[2]+1]}
if (horse_move1f=="") {horse_move1f<-board2[p_horse1[1]+2,p_horse1[2]+1]}
if (horse_move1g=="") {horse_move1g<-board2[p_horse1[1]+2,p_horse1[2]-1]}
if (horse_move1h=="") {horse_move1h<-board2[p_horse1[1]+1,p_horse1[2]-2]}}
if (p_horse1[2]==8 && p_horse1[1]>2 && p_horse1[1]<7){
  horse_move1a<-board[p_horse1[1]-1,p_horse1[2]-2]
  horse_move1b<-board[p_horse1[1]-2,p_horse1[2]-1]
  horse_move1g<-board[p_horse1[1]+2,p_horse1[2]-1]
  horse_move1h<-board[p_horse1[1]+1,p_horse1[2]-2]
  if (horse_move1a=="") {horse_move1a<-board2[p_horse1[1]-1,p_horse1[2]-2]}
  if (horse_move1b=="") {horse_move1b<-board2[p_horse1[1]-2,p_horse1[2]-1]}
  if (horse_move1g=="") {horse_move1g<-board2[p_horse1[1]+2,p_horse1[2]-1]}
  if (horse_move1h=="") {horse_move1h<-board2[p_horse1[1]+1,p_horse1[2]-2]}}
if (p_horse1[1]==1 && p_horse1[2]==1){
  horse_move1e<-board[p_horse1[1]+1,p_horse1[2]+2]
  horse_move1f<-board[p_horse1[1]+2,p_horse1[2]+1]
  if (horse_move1e=="") {horse_move1e<-board2[p_horse1[1]+1,p_horse1[2]+2]}
  if (horse_move1f=="") {horse_move1f<-board2[p_horse1[1]+2,p_horse1[2]+1]}}
if (p_horse1[1]==1 && p_horse1[2]==2){
  horse_move1e<-board[p_horse1[1]+1,p_horse1[2]+2]
  horse_move1f<-board[p_horse1[1]+2,p_horse1[2]+1]
  horse_move1g<-board[p_horse1[1]+2,p_horse1[2]-1]
  if (horse_move1e=="") {horse_move1e<-board2[p_horse1[1]+1,p_horse1[2]+2]}
  if (horse_move1f=="") {horse_move1f<-board2[p_horse1[1]+2,p_horse1[2]+1]}
  if (horse_move1g=="") {horse_move1g<-board2[p_horse1[1]+2,p_horse1[2]-1]}}
if (p_horse1[1]==1 && p_horse1[2]==7){
  horse_move1f<-board[p_horse1[1]+2,p_horse1[2]+1]
  horse_move1g<-board[p_horse1[1]+2,p_horse1[2]-1]
  horse_move1h<-board[p_horse1[1]+1,p_horse1[2]-2]
  if (horse_move1f=="") {horse_move1f<-board2[p_horse1[1]+2,p_horse1[2]+1]}
  if (horse_move1g=="") {horse_move1g<-board2[p_horse1[1]+2,p_horse1[2]-1]}
  if (horse_move1h=="") {horse_move1h<-board2[p_horse1[1]+1,p_horse1[2]-2]}}
if (p_horse1[1]==1 && p_horse1[2]==8){
  horse_move1g<-board[p_horse1[1]+2,p_horse1[2]-1]
  horse_move1h<-board[p_horse1[1]+1,p_horse1[2]-2]
  if (horse_move1g=="") {horse_move1g<-board2[p_horse1[1]+2,p_horse1[2]-1]}
  if (horse_move1h=="") {horse_move1h<-board2[p_horse1[1]+1,p_horse1[2]-2]}}
if (p_horse1[1]==2 && p_horse1[2]==1){
  horse_move1d<-board[p_horse1[1]-1,p_horse1[2]+2]
  horse_move1e<-board[p_horse1[1]+1,p_horse1[2]+2]
  horse_move1f<-board[p_horse1[1]+2,p_horse1[2]+1]
  if (horse_move1d=="") {horse_move1d<-board2[p_horse1[1]-1,p_horse1[2]+2]}
  if (horse_move1e=="") {horse_move1e<-board2[p_horse1[1]+1,p_horse1[2]+2]}
  if (horse_move1f=="") {horse_move1f<-board2[p_horse1[1]+2,p_horse1[2]+1]}}
if (p_horse1[1]==2 && p_horse1[2]==2){
  horse_move1d<-board[p_horse1[1]-1,p_horse1[2]+2]
  horse_move1e<-board[p_horse1[1]+1,p_horse1[2]+2]
  horse_move1f<-board[p_horse1[1]+2,p_horse1[2]+1]
  horse_move1g<-board[p_horse1[1]+2,p_horse1[2]-1]
  if (horse_move1d=="") {horse_move1d<-board2[p_horse1[1]-1,p_horse1[2]+2]}
  if (horse_move1e=="") {horse_move1e<-board2[p_horse1[1]+1,p_horse1[2]+2]}
  if (horse_move1f=="") {horse_move1f<-board2[p_horse1[1]+2,p_horse1[2]+1]}}

```



```

    if (horse_move1d=="") {horse_move1c<-board2[p_horse1[1]-1,p_horse1[2]+2]}}
if (p_horse1[1]==8 && p_horse1[2]==2){
  horse_move1b<-board[p_horse1[1]-2,p_horse1[2]-1]
  horse_move1c<-board[p_horse1[1]-2,p_horse1[2]+1]
  horse_move1d<-board[p_horse1[1]-1,p_horse1[2]+2]
  if (horse_move1b=="") {horse_move1b<-board2[p_horse1[1]-2,p_horse1[2]-1]}
  if (horse_move1c=="") {horse_move1c<-board2[p_horse1[1]-2,p_horse1[2]+1]}
  if (horse_move1d=="") {horse_move1d<-board2[p_horse1[1]-1,p_horse1[2]+2]}}
if (p_horse1[1]==8 && p_horse1[2]==7){
  horse_move1a<-board[p_horse1[1]-1,p_horse1[2]-2]
  horse_move1b<-board[p_horse1[1]-2,p_horse1[2]-1]
  horse_move1c<-board[p_horse1[1]-2,p_horse1[2]+1]
  if (horse_move1a=="") {horse_move1a<-board2[p_horse1[1]-1,p_horse1[2]-2]}
  if (horse_move1b=="") {horse_move1b<-board2[p_horse1[1]-2,p_horse1[2]-1]}
  if (horse_move1c=="") {horse_move1c<-board2[p_horse1[1]-2,p_horse1[2]+1]}}
if (p_horse1[1]==8 && p_horse1[2]==8){
  horse_move1a<-board[p_horse1[1]-1,p_horse1[2]-2]
  horse_move1b<-board[p_horse1[1]-2,p_horse1[2]-1]
  if (horse_move1a=="") {horse_move1a<-board2[p_horse1[1]-1,p_horse1[2]-2]}
  if (horse_move1b=="") {horse_move1b<-board2[p_horse1[1]-2,p_horse1[2]-1]}}
if (all(color1==color2)==FALSE){
  if (horse_move1a %in% color1 == TRUE | horse_move1a=="sample") {horse_move1a<-numeric()}
  if (horse_move1b %in% color1 == TRUE | horse_move1b=="sample") {horse_move1b<-numeric()}
  if (horse_move1c %in% color1 == TRUE | horse_move1c=="sample") {horse_move1c<-numeric()}
  if (horse_move1d %in% color1 == TRUE | horse_move1d=="sample") {horse_move1d<-numeric()}
  if (horse_move1e %in% color1 == TRUE | horse_move1e=="sample") {horse_move1e<-numeric()}
  if (horse_move1f %in% color1 == TRUE | horse_move1f=="sample") {horse_move1f<-numeric()}
  if (horse_move1g %in% color1 == TRUE | horse_move1g=="sample") {horse_move1g<-numeric()}
  if (horse_move1h %in% color1 == TRUE | horse_move1h=="sample") {horse_move1h<-numeric()}}
if (all(color1==color2)==TRUE){
  if (horse_move1a=="sample") {horse_move1a<-numeric()}
  if (horse_move1b=="sample") {horse_move1b<-numeric()}
  if (horse_move1c=="sample") {horse_move1c<-numeric()}
  if (horse_move1d=="sample") {horse_move1d<-numeric()}
  if (horse_move1e=="sample") {horse_move1e<-numeric()}
  if (horse_move1f=="sample") {horse_move1f<-numeric()}
  if (horse_move1g=="sample") {horse_move1g<-numeric()}
  if (horse_move1h=="sample") {horse_move1h<-numeric()}}
horse1_moves<-c(horse_move1a,horse_move1b,horse_move1c,horse_move1d,
                 horse_move1e,horse_move1f,horse_move1g,horse_move1h)
return(horse1_moves)}

#white pawn
white_pawn<-function(board,p_pawn1){
if ((p_pawn1[1]-1)>0){
  if ((p_pawn1[2]-1)<7 && (p_pawn1[2]+1)>2){
    pawn_move1a<-board[p_pawn1[1]-1,p_pawn1[2]]
    if (pawn_move1a=="") {pawn_move1a=board2[p_pawn1[1]-1,p_pawn1[2]]}
    if (p_pawn1[1]==7){
      pawn_move1a<-append(pawn_move1a,board[5,p_pawn1[2]])
      if (pawn_move1a[2]=="") {pawn_move1a[2]=board2[5,p_pawn1[2]]}
      if (pawn_move1a[2] %in% black == TRUE | pawn_move1a[2] %in% white == TRUE){
        pawn_move1a<-pawn_move1a[1]}}}
```

```

if (pawn_move1a[1] %in% black == TRUE||pawn_move1a[1] %in% white == TRUE){pawn_move1a<-numeric()}
pawn_move1b<-board[p_pawn1[1]-1,p_pawn1[2]-1]
if (pawn_move1b==""|pawn_move1b %in% white == TRUE){pawn_move1b<-numeric()}
pawn_move1c<-board[p_pawn1[1]-1,p_pawn1[2]+1]
if (pawn_move1c==""|pawn_move1c %in% white == TRUE){pawn_move1c<-numeric()}}
if (p_pawn1[2]==1){
  pawn_move1a<-board[p_pawn1[1]-1,1]
  if (pawn_move1a==""){pawn_move1a=board2[p_pawn1[1]-1,1]}
  if (p_pawn1[1]==7){
    pawn_move1a<-append(pawn_move1a,board[5,1])
    if (pawn_move1a[2]== ""){pawn_move1a[2]=board2[5,1]}
    if (pawn_move1a[2] %in% black == TRUE||pawn_move1a[2] %in% white == TRUE){
      pawn_move1a<-pawn_move1a[1]}
  if (pawn_move1a[1] %in% black == TRUE||pawn_move1a[1] %in% white == TRUE){pawn_move1a<-numeric()}
  pawn_move1b<-numeric()
  pawn_move1c<-board[p_pawn1[1]-1,2]
  if (pawn_move1c==""|pawn_move1c %in% white == TRUE){pawn_move1c<-numeric()}}
if (p_pawn1[2]==8){
  pawn_move1a<-board[p_pawn1[1]-1,8]
  if (pawn_move1a==""){pawn_move1a=board2[p_pawn1[1]-1,8]}
  if (p_pawn1[1]==7){
    pawn_move1a<-append(pawn_move1a,board[5,8])
    if (pawn_move1a[2]== ""){pawn_move1a[2]=board2[5,8]}
    if (pawn_move1a[2] %in% black == TRUE||pawn_move1a[2] %in% white == TRUE){
      pawn_move1a<-pawn_move1a[1]}
  if (pawn_move1a[1] %in% black == TRUE||pawn_move1a[1] %in% white == TRUE){pawn_move1a<-numeric()}
  pawn_move1b<-board[p_pawn1[1]-1,7]
  if (pawn_move1b==""|pawn_move1b %in% white == TRUE){pawn_move1b<-numeric()}
  pawn_move1c<-numeric()}}
f_pawn_move1<-c(pawn_move1a,pawn_move1b,pawn_move1c)}
if ((p_pawn1[1]-1)==0){f_pawn_move1<-numeric()}
return(f_pawn_move1)

#black pawn
black_pawn<-function(board,p_pawn1){
if ((p_pawn1[1]+1)<9){
  if ((p_pawn1[2]-1)<7 && (p_pawn1[2]+1)>2){
    pawn_move1a<-board[p_pawn1[1]+1,p_pawn1[2]]
    if (pawn_move1a==""){pawn_move1a=board2[p_pawn1[1]+1,p_pawn1[2]]}
    if (p_pawn1[1]==2){
      pawn_move1a<-append(pawn_move1a,board[4,p_pawn1[2]])
      if (pawn_move1a[2]== ""){pawn_move1a[2]=board2[4,p_pawn1[2]]}
      if (pawn_move1a[2] %in% black == TRUE||pawn_move1a[2] %in% white == TRUE){
        pawn_move1a<-pawn_move1a[1]}
    if (pawn_move1a[1] %in% black == TRUE||pawn_move1a[1] %in% white == TRUE){pawn_move1a<-numeric()}
    pawn_move1b<-board[p_pawn1[1]+1,p_pawn1[2]-1]
    if (pawn_move1b==""|pawn_move1b %in% black == TRUE){pawn_move1b<-numeric()}
    pawn_move1c<-board[p_pawn1[1]+1,p_pawn1[2]+1]
    if (pawn_move1c==""|pawn_move1c %in% black == TRUE){pawn_move1c<-numeric()}}
  if (p_pawn1[2]==1{
    pawn_move1a<-board[p_pawn1[1]+1,1]
    if (pawn_move1a==""){pawn_move1a=board2[p_pawn1[1]+1,1]}
    if (p_pawn1[1]==2){

```

```

pawn_move1a<-append(pawn_move1a,board[4,1])
if (pawn_move1a[2]== ""){pawn_move1a[2]=board2[4,1]}
if (pawn_move1a[2] %in% black == TRUE|pawn_move1a[2] %in% white == TRUE){
    pawn_move1a<-pawn_move1a[1]}
if (pawn_move1a[1] %in% black == TRUE|pawn_move1a[1] %in% white == TRUE){pawn_move1a<-numeric()}
pawn_move1b<-numeric()
pawn_move1c<-board[p_pawn1[1]+1,2]
if (pawn_move1c== ""|pawn_move1c %in% black == TRUE){pawn_move1c<-numeric()}
if (p_pawn1[2]==8){
    pawn_move1a<-board[p_pawn1[1]+1,8]
    if (pawn_move1a== ""){pawn_move1a=board2[p_pawn1[1]+1,8]}
    if (p_pawn1[1]==2){
        pawn_move1a<-append(pawn_move1a,board[4,8])
        if (pawn_move1a[2]== ""){pawn_move1a[2]=board2[4,8]}
        if (pawn_move1a[2] %in% black == TRUE|pawn_move1a[2] %in% white == TRUE){
            pawn_move1a<-pawn_move1a[1]}
        if (pawn_move1a[1] %in% black == TRUE|pawn_move1a[1] %in% white == TRUE){pawn_move1a<-numeric()}
        pawn_move1b<-board[p_pawn1[1]+1,7]
        if (pawn_move1b== ""|pawn_move1b %in% black == TRUE){pawn_move1b<-numeric()}
        pawn_move1c<-numeric()
f_pawn_move1<-c(pawn_move1a,pawn_move1b,pawn_move1c)}
if ((p_pawn1[1]+1)==9){f_pawn_move1<-numeric()}
return(f_pawn_move1}

#pawn going up
white_opponent<-function(board,p_pawn1,color1,color2){
if ((p_pawn1[1]-1)>0){
    if ((p_pawn1[2]-1)<7 && (p_pawn1[2]+1)>2){
        pawn_move1b<-board[p_pawn1[1]-1,p_pawn1[2]-1]
        if (pawn_move1b== ""){pawn_move1b<-board2[p_pawn1[1]-1,p_pawn1[2]-1]}
        if (all(color1==color2)==FALSE){
            if (pawn_move1b %in% color1 == TRUE){pawn_move1b<-numeric()}}
        pawn_move1c<-board[p_pawn1[1]-1,p_pawn1[2]+1]
        if (pawn_move1c== ""){pawn_move1c<-board2[p_pawn1[1]-1,p_pawn1[2]+1]}
        if (all(color1==color2)==FALSE){
            if (pawn_move1c %in% color1 == TRUE){pawn_move1c<-numeric()}}
        if (p_pawn1[2]==1){
            pawn_move1b<-numeric()
            pawn_move1c<-board[p_pawn1[1]-1,2]
            if (pawn_move1c== ""){pawn_move1c<-board2[p_pawn1[1]-1,2]}
            if (all(color1==color2)==FALSE){
                if (pawn_move1c %in% color1 == TRUE){pawn_move1c<-numeric()}}
        if (p_pawn1[2]==8){
            pawn_move1b<-board[p_pawn1[1]-1,7]
            if (pawn_move1b== ""){pawn_move1b<-board2[p_pawn1[1]-1,7]}
            if (all(color1==color2)==FALSE){
                if (pawn_move1b %in% color1 == TRUE){pawn_move1b<-numeric()}}
            pawn_move1c<-numeric()
f_pawn_move1<-c(pawn_move1b,pawn_move1c)}
if ((p_pawn1[1]-1)==0){f_pawn_move1<-numeric()}
return(f_pawn_move1}

#pawn going down

```

```

pawn_opponent<-function(board,p_pawn1,color1,color2){
if ((p_pawn1[1]+1)<9){
  if ((p_pawn1[2]-1)<7 && (p_pawn1[2]+1)>2){
    pawn_move1b<-board[p_pawn1[1]+1,p_pawn1[2]-1]
    if (pawn_move1b==""){pawn_move1b<-board2[p_pawn1[1]+1,p_pawn1[2]-1]}
    if (all(color1==color2)==FALSE){
      if (pawn_move1b %in% color1 == TRUE){pawn_move1b<-numeric()}}
    pawn_move1c<-board[p_pawn1[1]+1,p_pawn1[2]+1]
    if (pawn_move1c==""){pawn_move1c<-board2[p_pawn1[1]+1,p_pawn1[2]+1]}
    if (all(color1==color2)==FALSE){
      if (pawn_move1c %in% color1 == TRUE){pawn_move1c<-numeric()}}
    if (p_pawn1[2]==1){
      pawn_move1b<-numeric()
      pawn_move1c<-board[p_pawn1[1]+1,2]
      if (pawn_move1c==""){pawn_move1c<-board2[p_pawn1[1]+1,2]}
      if (all(color1==color2)==FALSE){
        if (pawn_move1c %in% color1 == TRUE){pawn_move1c<-numeric()}}
    if (p_pawn1[2]==8){
      pawn_move1b<-board[p_pawn1[1]+1,7]
      if (pawn_move1b==""){pawn_move1b<-board2[p_pawn1[1]+1,7]}
      if (all(color1==color2)==FALSE){
        if (pawn_move1b %in% color1 == TRUE){pawn_move1b<-numeric()}}
      pawn_move1c<-numeric()
    f_pawn_move1<-c(pawn_move1b,pawn_move1c)}
  if ((p_pawn1[1]+1)==9){f_pawn_move1<-numeric()}
  return(f_pawn_move1)}

#select play as white
#black computer move, 'from' and 'to' are located from the user interface
black_move<-function(from,to,chessboard){
  board <- move_piece(chessboard,from,to)

#white move
  mking<-matrix(nrow=0,ncol=2)
  mqueen<-matrix(nrow=0,ncol=2)
  mrook1<-matrix(nrow=0,ncol=2)
  mrook2<-matrix(nrow=0,ncol=2)
  mhorse1<-matrix(nrow=0,ncol=2)
  mhorse2<-matrix(nrow=0,ncol=2)
  mbishop1<-matrix(nrow=0,ncol=2)
  mbishop2<-matrix(nrow=0,ncol=2)
  mp1<-matrix(nrow=0,ncol=2)
  mp2<-matrix(nrow=0,ncol=2)
  mp3<-matrix(nrow=0,ncol=2)
  mp4<-matrix(nrow=0,ncol=2)
  mp5<-matrix(nrow=0,ncol=2)
  mp6<-matrix(nrow=0,ncol=2)
  mp7<-matrix(nrow=0,ncol=2)
  mp8<-matrix(nrow=0,ncol=2)
  b_king<-which(board=="k", arr.ind=TRUE)
  mix_king<-king(board,b_king,black,black)
  mking<-matrix(nrow=length(mix_king),ncol=2)
  if (length(mix_king)>0){
    for (i in 1:length(mix_king)){

```

```

mking[i,1]="k"
mking[i,2]=mix_king[i]}
if ("p_1" %in% board){
  p_pawn1<-which(board=="p_1", arr.ind=TRUE)
  white_move<-white_opponent(board,p_pawn1,black,black)
  mp1<-matrix(nrow=length(white_move),ncol=2)
  if (length(white_move)>0){
    for (i in 1:length(white_move)){
      mp1[i,1]="p_1"
      mp1[i,2]=white_move[i]}}
  if ("p_2" %in% board){
    p_pawn1<-which(board=="p_2", arr.ind=TRUE)
    white_move<-white_opponent(board,p_pawn1,black,black)
    mp2<-matrix(nrow=length(white_move),ncol=2)
    if (length(white_move)>0){
      for (i in 1:length(white_move)){
        mp2[i,1]="p_2"
        mp2[i,2]=white_move[i]}}
  if ("p_3" %in% board){
    p_pawn1<-which(board=="p_3", arr.ind=TRUE)
    white_move<-white_opponent(board,p_pawn1,black,black)
    mp3<-matrix(nrow=length(white_move),ncol=2)
    if (length(white_move)>0){
      for (i in 1:length(white_move)){
        mp3[i,1]="p_3"
        mp3[i,2]=white_move[i]}}
  if ("p_4" %in% board){
    p_pawn1<-which(board=="p_4", arr.ind=TRUE)
    white_move<-white_opponent(board,p_pawn1,black,black)
    mp4<-matrix(nrow=length(white_move),ncol=2)
    if (length(white_move)>0){
      for (i in 1:length(white_move)){
        mp4[i,1]="p_4"
        mp4[i,2]=white_move[i]}}
  if ("p_5" %in% board){
    p_pawn1<-which(board=="p_5", arr.ind=TRUE)
    white_move<-white_opponent(board,p_pawn1,black,black)
    mp5<-matrix(nrow=length(white_move),ncol=2)
    if (length(white_move)>0){
      for (i in 1:length(white_move)){
        mp5[i,1]="p_5"
        mp5[i,2]=white_move[i]}}
  if ("p_6" %in% board){
    p_pawn1<-which(board=="p_6", arr.ind=TRUE)
    white_move<-white_opponent(board,p_pawn1,black,black)
    mp6<-matrix(nrow=length(white_move),ncol=2)
    if (length(white_move)>0){
      for (i in 1:length(white_move)){
        mp6[i,1]="p_6"
        mp6[i,2]=white_move[i]}}
  if ("p_7" %in% board){
    p_pawn1<-which(board=="p_7", arr.ind=TRUE)
    white_move<-white_opponent(board,p_pawn1,black,black)

```

```

mp7<-matrix(nrow=length(white_move) ,ncol=2)
if (length(white_move)>0){
for (i in 1:length(white_move)){
  mp7[i,1]="p_7"
  mp7[i,2]=white_move[i]}}
if ("p_8" %in% board){
p_pawn1<-which(board=="p_8", arr.ind=TRUE)
white_move<-white_opponent(board,p_pawn1,black,black)
mp8<-matrix(nrow=length(white_move) ,ncol=2)
if (length(white_move)>0){
for (i in 1:length(white_move)){
  mp8[i,1]="p_8"
  mp8[i,2]=white_move[i]}}
if ("r_1" %in% board){
p_rook1<-which(board=="r_1", arr.ind=TRUE)
white_move<-rook(board,p_rook1,black,black)
mrook1<-matrix(nrow=length(white_move) ,ncol=2)
if (length(white_move)>0){
for (i in 1:length(white_move)){
  mrook1[i,1]="r_1"
  mrook1[i,2]=white_move[i]}}
if ("r_2" %in% board){
p_rook1<-which(board=="r_2", arr.ind=TRUE)
white_move<-rook(board,p_rook1,black,black)
mrook2<-matrix(nrow=length(white_move) ,ncol=2)
if (length(white_move)>0){
for (i in 1:length(white_move)){
  mrook2[i,1]="r_2"
  mrook2[i,2]=white_move[i]}}
if ("h_1" %in% board){
p_horse1<-which(board=="h_1", arr.ind=TRUE)
white_move<-horse(board,p_horse1,black,black)
mhorse1<-matrix(nrow=length(white_move) ,ncol=2)
if (length(white_move)>0){
for (i in 1:length(white_move)){
  mhorse1[i,1]="h_1"
  mhorse1[i,2]=white_move[i]}}
if ("h_2" %in% board){
p_horse1<-which(board=="h_2", arr.ind=TRUE)
white_move<-horse(board,p_horse1,black,black)
mhorse2<-matrix(nrow=length(white_move) ,ncol=2)
if (length(white_move)>0){
for (i in 1:length(white_move)){
  mhorse2[i,1]="h_2"
  mhorse2[i,2]=white_move[i]}}
if ("b_1" %in% board){
p_bishop1<-which(board=="b_1", arr.ind=TRUE)
white_move<-bishop(board,p_bishop1,black,black)
mbishop1<-matrix(nrow=length(white_move) ,ncol=2)
if (length(white_move)>0){
for (i in 1:length(white_move)){
  mbishop1[i,1]="b_1"
  mbishop1[i,2]=white_move[i]}}

```

```

if ("b_2" %in% board){
  p_bishop1<-which(board=="b_2", arr.ind=TRUE)
  white_move<-bishop(board,p_bishop1,black,black)
  mbishop2<-matrix(nrow=length(white_move),ncol=2)
  if (length(white_move)>0){
    for (i in 1:length(white_move)){
      mbishop2[i,1]="b_2"
      mbishop2[i,2]=white_move[i]}}}
if ("q" %in% board){
  p_queen<-which(board=="q", arr.ind=TRUE)
  white_move<-queen(board,p_queen,black,black)
  mqueen<-matrix(nrow=length(white_move),ncol=2)
  if (length(white_move)>0){
    for (i in 1:length(white_move)){
      mqueen[i,1]="q"
      mqueen[i,2]=white_move[i]}}}
score_white<-do.call(rbind,list(mking,mqueen,mrook1,mrook2,mhorse1,mhorse2,mbishop1,mbishop2,
                                 mp1,mp2,mp3,mp4,mp5,mp6,mp7,mp8))
white_move<-score_white[,2]
white_move<-unique(white_move)

#check
p_king<-which(board=="k", arr.ind=TRUE)
target2<-king(board,p_king,white,black)
target<-append("k",target2)
same<-Reduce(intersect, list(black,target))
changed<-numeric()
if (length(same)>0){
  for (i in 1:length(same)){
    changes=same[i]
    change<-which(board==changes, arr.ind=TRUE)
    changing<-board2[change[1],change[2]]
    board[change[1],change[2]]=changing
    changed<-append(changed,changing)}}

#black move
P8_listing<-list()
P7_listing<-list()
P6_listing<-list()
P5_listing<-list()
P4_listing<-list()
P3_listing<-list()
P2_listing<-list()
P1_listing<-list()
b2_listing<-list()
b1_listing<-list()
h2_listing<-list()
h1_listing<-list()
r2_listing<-list()
r1_listing<-list()
q_listing<-list()
k_listing<-list()
group<-numeric()

```

```

if ("P_8" %in% board){
  p_pawn1<-which(board=="P_8", arr.ind=TRUE)
  BP8_move<-black_pawn(board,p_pawn1)
  if (length(changed)>0){
    BP8_move<-BP8_move[BP8_move!=changed]
    BP8_move<-BP8_move[!is.na(BP8_move)]}
  initial<-BP8_move
  if (length(BP8_move)>0){
    l=length(BP8_move)
    for (j in 1:l){
      mat<-matrix(nrow=48,ncol=49)
      mat[,1]<-c(rep("P_8",48))
      mat[,2]<-c(rep(initial[j],48))
      compare=FALSE
      BP8_move<-initial[j]
      high=0
      stop=c(1,1)
      while (compare==FALSE){
        stop<-append(stop,1)
        s<-length(stop)
        cap1<-numeric()
        cap2<-numeric()
        for (i in 1:length(BP8_move)){
          if (BP8_move[i] %in% board2){position<-which(board2==BP8_move[i], arr.ind=TRUE)}
          if (BP8_move[i] %in% board){position<-which(board==BP8_move[i], arr.ind=TRUE)}
          cap1_horse<-black_pawn(board,position)
          cap1<-append(cap1,cap1_horse)
          cap1<-unique(cap1)
          cap2_pawn<-pawn_opponent(board,position,black,white)
          cap2<-append(cap2,cap2_pawn)
          cap2<-unique(cap2)}
        maximum<-length(cap2)
        high<-max(high,maximum)
        BP8_move<-cap1
        cap3<-append(cap2,numERIC(48-maximum))
        if (maximum>0){mat[,s]<-cap3}
        if (is.character(cap1)==FALSE){break}}
      mat2<-mat[1:high,]
      dm<-dim(mat2)
      dn<-dm[1]
      if (is.null(dm)==TRUE){dn=1}
      group<-append(group,dn)
      P8_listing<-append(P8_listing,list(mat2))}}
  if ("P_7" %in% board){
    p_pawn1<-which(board=="P_7", arr.ind=TRUE)
    BP7_move<-black_pawn(board,p_pawn1)
    if (length(changed)>0){
      BP7_move<-BP7_move[BP7_move!=changed]
      BP7_move<-BP7_move[!is.na(BP7_move)]}
    initial<-BP7_move
    if (length(BP7_move)>0){
      l=length(BP7_move)
      for (j in 1:l){

```

```

mat<-matrix(nrow=48,ncol=49)
mat[,1]<-c(rep("P_7",48))
mat[,2]<-c(rep(initial[j],48))
compare=FALSE
BP7_move<-initial[j]
high=0
stop=c(1,1)
while (compare==FALSE){
  stop<-append(stop,1)
  s<-length(stop)
  cap1<-numeric()
  cap2<-numeric()
  for (i in 1:length(BP7_move)){
    if (BP7_move[i] %in% board2){position<-which(board2==BP7_move[i], arr.ind=TRUE)}
    if (BP7_move[i] %in% board){position<-which(board==BP7_move[i], arr.ind=TRUE)}
    cap1_horse<-black_pawn(board,position)
    cap1<-append(cap1,cap1_horse)
    cap1<-unique(cap1)
    cap2_pawn<-pawn_opponent(board,position,black,white)
    cap2<-append(cap2,cap2_pawn)
    cap2<-unique(cap2)}
    maximum<-length(cap2)
    high<-max(high,maximum)
    BP7_move<-cap1
    cap3<-append(cap2,numeric(48-maximum))
    if (maximum>0){mat[,s]<-cap3}
    if (is.character(cap1)==FALSE){break}}
  mat2<-mat[1:high,]
  dm<-dim(mat2)
  dn<-dm[1]
  if (is.null(dm)==TRUE){dn=1}
  group<-append(group,dn)
  P7_listing<-append(P7_listing,list(mat2))}}
if ("P_6" %in% board){
  p_pawn1<-which(board=="P_6", arr.ind=TRUE)
  BP6_move<-black_pawn(board,p_pawn1)
  if (length(changed)>0){
    BP6_move<-BP6_move[BP6_move!=changed]
    BP6_move<-BP6_move[!is.na(BP6_move)]}
  initial<-BP6_move
  if (length(BP6_move)>0){
    l=length(BP6_move)
    for (j in 1:l){
      mat<-matrix(nrow=48,ncol=49)
      mat[,1]<-c(rep("P_6",48))
      mat[,2]<-c(rep(initial[j],48))
      compare=FALSE
      BP6_move<-initial[j]
      high=0
      stop=c(1,1)
      while (compare==FALSE){
        stop<-append(stop,1)
        s<-length(stop)

```

```

cap1<-numeric()
cap2<-numeric()
for (i in 1:length(BP6_move)){
  if (BP6_move[i] %in% board2){position<-which(board2==BP6_move[i], arr.ind=TRUE)}
  if (BP6_move[i] %in% board){position<-which(board==BP6_move[i], arr.ind=TRUE)}
  cap1_horse<-black_pawn(board,position)
  cap1<-append(cap1,cap1_horse)
  cap1<-unique(cap1)
  cap2_pawn<-pawn_opponent(board,position,black,white)
  cap2<-append(cap2,cap2_pawn)
  cap2<-unique(cap2)}
  maximum<-length(cap2)
  high<-max(high,maximum)
  BP6_move<-cap1
  cap3<-append(cap2,numERIC(48-maximum))
  if (maximum>0){mat[,s]<-cap3}
  if (is.character(cap1)==FALSE){break}}
  mat2<-mat[1:high,]
  dm<-dim(mat2)
  dn<-dm[1]
  if (is.null(dm)==TRUE){dn=1}
  group<-append(group,dn)
  P6_listing<-append(P6_listing,list(mat2))}}
if ("P_5" %in% board){
  p_pawn1<-which(board=="P_5", arr.ind=TRUE)
  BP5_move<-black_pawn(board,p_pawn1)
  if (length(changed)>0){
    BP5_move<-BP5_move[BP5_move!=changed]
    BP5_move<-BP5_move[!is.na(BP5_move)]}
  initial<-BP5_move
  if (length(BP5_move)>0){
    l=length(BP5_move)
    for (j in 1:l){
      mat<-matrix(nrow=48,ncol=49)
      mat[,1]<-c(rep("P_5",48))
      mat[,2]<-c(rep(initial[j],48))
      compare=FALSE
      BP5_move<-initial[j]
      high=0
      stop=c(1,1)
      while (compare==FALSE){
        stop<-append(stop,1)
        s<-length(stop)
        cap1<-numeric()
        cap2<-numeric()
        for (i in 1:length(BP5_move)){
          if (BP5_move[i] %in% board2){position<-which(board2==BP5_move[i], arr.ind=TRUE)}
          if (BP5_move[i] %in% board){position<-which(board==BP5_move[i], arr.ind=TRUE)}
          cap1_horse<-black_pawn(board,position)
          cap1<-append(cap1,cap1_horse)
          cap1<-unique(cap1)
          cap2_pawn<-pawn_opponent(board,position,black,white)
          cap2<-append(cap2,cap2_pawn)

```

```

        cap2<-unique(cap2)}
maximum<-length(cap2)
high<-max(high,maximum)
BP5_move<-cap1
cap3<-append(cap2,numeric(48-maximum))
if (maximum>0){mat[,s]<-cap3}
if (is.character(cap1)==FALSE){break}}
mat2<-mat[1:high,]
dm<-dim(mat2)
dn<-dm[1]
if (is.null(dm)==TRUE){dn=1}
group<-append(group,dn)
P5_listing<-append(P5_listing,list(mat2))}}
if ("P_4" %in% board){
p_pawn1<-which(board=="P_4", arr.ind=TRUE)
BP4_move<-black_pawn(board,p_pawn1)
if (length(changed)>0){
    BP4_move<-BP4_move[BP4_move!=changed]
    BP4_move<-BP4_move[!is.na(BP4_move)]}
initial<-BP4_move
if (length(BP4_move)>0){
    l=length(BP4_move)
    for (j in 1:l){
        mat<-matrix(nrow=48,ncol=49)
        mat[,1]<-c(rep("P_4",48))
        mat[,2]<-c(rep(initial[j],48))
        compare=FALSE
        BP4_move<-initial[j]
        high=0
        stop=c(1,1)
        while (compare==FALSE){
            stop<-append(stop,1)
            s<-length(stop)
            cap1<-numeric()
            cap2<-numeric()
            for (i in 1:length(BP4_move)){
                if (BP4_move[i] %in% board2){position<-which(board2==BP4_move[i], arr.ind=TRUE)}
                if (BP4_move[i] %in% board){position<-which(board==BP4_move[i], arr.ind=TRUE)}
                cap1_horse<-black_pawn(board,position)
                cap1<-append(cap1,cap1_horse)
                cap1<-unique(cap1)
                cap2_pawn<-pawn_opponent(board,position,black,white)
                cap2<-append(cap2,cap2_pawn)
                cap2<-unique(cap2)}
            maximum<-length(cap2)
            high<-max(high,maximum)
            BP4_move<-cap1
            cap3<-append(cap2,numeric(48-maximum))
            if (maximum>0){mat[,s]<-cap3}
            if (is.character(cap1)==FALSE){break}}
        mat2<-mat[1:high,]
        dm<-dim(mat2)
        dn<-dm[1]
    }
}
}

```

```

    if (is.null(dm)==TRUE){dn=1}
    group<-append(group,dn)
    P4_listing<-append(P4_listing,list(mat2))}}
if ("P_3" %in% board){
  p_pawn1<-which(board=="P_3", arr.ind=TRUE)
  BP3_move<-black_pawn(board,p_pawn1)
  if (length(changed)>0){
    BP3_move<-BP3_move[BP3_move!=changed]
    BP3_move<-BP3_move[!is.na(BP3_move)]}
  initial<-BP3_move
  if (length(BP3_move)>0){
    l=length(BP3_move)
    for (j in 1:l){
      mat<-matrix(nrow=48,ncol=49)
      mat[,1]<-c(rep("P_3",48))
      mat[,2]<-c(rep(initial[j],48))
      compare=FALSE
      BP3_move<-initial[j]
      high=0
      stop=c(1,1)
      while (compare==FALSE){
        stop<-append(stop,1)
        s<-length(stop)
        cap1<-numeric()
        cap2<-numeric()
        for (i in 1:length(BP3_move)){
          if (BP3_move[i] %in% board2){position<-which(board2==BP3_move[i], arr.ind=TRUE)}
          if (BP3_move[i] %in% board){position<-which(board==BP3_move[i], arr.ind=TRUE)}
          cap1_horse<-black_pawn(board,position)
          cap1<-append(cap1,cap1_horse)
          cap1<-unique(cap1)
          cap2_pawn<-pawn_opponent(board,position,black,white)
          cap2<-append(cap2,cap2_pawn)
          cap2<-unique(cap2)}
        maximum<-length(cap2)
        high<-max(high,maximum)
        BP3_move<-cap1
        cap3<-append(cap2,numeric(48-maximum))
        if (maximum>0){mat[,s]<-cap3}
        if (is.character(cap1)==FALSE){break}}
      mat2<-mat[1:high,]
      dm<-dim(mat2)
      dn<-dm[1]
      if (is.null(dm)==TRUE){dn=1}
      group<-append(group,dn)
      P3_listing<-append(P3_listing,list(mat2))}}
  if ("P_2" %in% board){
    p_pawn1<-which(board=="P_2", arr.ind=TRUE)
    BP2_move<-black_pawn(board,p_pawn1)
    if (length(changed)>0){
      BP2_move<-BP2_move[BP2_move!=changed]
      BP2_move<-BP2_move[!is.na(BP2_move)]}
    initial<-BP2_move

```

```

if (length(BP2_move)>0){
  l=length(BP2_move)
  for (j in 1:l){
    mat<-matrix(nrow=48,ncol=49)
    mat[,1]<-c(rep("P_2",48))
    mat[,2]<-c(rep(initial[j],48))
    compare=FALSE
    BP2_move<-initial[j]
    high=0
    stop=c(1,1)
    while (compare==FALSE){
      stop<-append(stop,1)
      s<-length(stop)
      cap1<-numeric()
      cap2<-numeric()
      for (i in 1:length(BP2_move)){
        if (BP2_move[i] %in% board2){position<-which(board2==BP2_move[i], arr.ind=TRUE)}
        if (BP2_move[i] %in% board){position<-which(board==BP2_move[i], arr.ind=TRUE)}
        cap1_horse<-black_pawn(board,position)
        cap1<-append(cap1,cap1_horse)
        cap1<-unique(cap1)
        cap2_pawn<-pawn_opponent(board,position,black,white)
        cap2<-append(cap2,cap2_pawn)
        cap2<-unique(cap2)}
        maximum<-length(cap2)
        high<-max(high,maximum)
        BP2_move<-cap1
        cap3<-append(cap2,numERIC(48-maximum))
        if (maximum>0){mat[,s]<-cap3}
        if (is.character(cap1)==FALSE){break}}
      mat2<-mat[1:high,]
      dm<-dim(mat2)
      dn<-dm[1]
      if (is.null(dm)==TRUE){dn=1}
      group<-append(group,dn)
      P2_listing<-append(P2_listing,list(mat2))}}
  if ("P_1" %in% board){
    p_pawn1<-which(board=="P_1", arr.ind=TRUE)
    BP1_move<-black_pawn(board,p_pawn1)
    if (length(changed)>0){
      BP1_move<-BP1_move[BP1_move!=changed]
      BP1_move<-BP1_move[!is.na(BP1_move)]}
    initial<-BP1_move
    if (length(BP1_move)>0){
      l=length(BP1_move)
      for (j in 1:l){
        mat<-matrix(nrow=48,ncol=49)
        mat[,1]<-c(rep("P_1",48))
        mat[,2]<-c(rep(initial[j],48))
        compare=FALSE
        BP1_move<-initial[j]
        high=0
        stop=c(1,1)

```

```

while (compare==FALSE){
  stop<-append(stop,1)
  s<-length(stop)
  cap1<-numeric()
  cap2<-numeric()
  for (i in 1:length(BP1_move)){
    if (BP1_move[i] %in% board2){position<-which(board2==BP1_move[i], arr.ind=TRUE)}
    if (BP1_move[i] %in% board){position<-which(board==BP1_move[i], arr.ind=TRUE)}
    cap1_horse<-black_pawn(board,position)
    cap1<-append(cap1,cap1_horse)
    cap1<-unique(cap1)
    cap2_pawn<-pawn_opponent(board,position,black,white)
    cap2<-append(cap2,cap2_pawn)
    cap2<-unique(cap2)}
    maximum<-length(cap2)
    high<-max(high,maximum)
    BP1_move<-cap1
    cap3<-append(cap2,numeric(48-maximum))
    if (maximum>0){mat[,s]<-cap3}
    if (is.character(cap1)==FALSE){break}}
  mat2<-mat[1:high,]
  dm<-dim(mat2)
  dn<-dm[1]
  if (is.null(dm)==TRUE){dn=1}
  group<-append(group,dn)
  P1_listing<-append(P1_listing,list(mat2))}

#officials
if ("B_2" %in% board){
  p_bishop1<-which(board=="B_2", arr.ind=TRUE)
  bbishop2_move<-bishop(board,p_bishop1,black,white)
  if (length(changed)>0){
    bbishop2_move<-bbishop2_move[bbishop2_move!=changed]
    bbishop2_move<-bbishop2_move[!is.na(bbishop2_move)]}
  initial<-bbishop2_move
  if (length(bbishop2_move)>0){
    l=length(bbishop2_move)
    for (j in 1:l){
      mat<-matrix(nrow=62,ncol=49)
      mat[,1]<-c(rep("B_2",62))
      mat[,2]<-c(rep(initial[j],62))
      bbishop2_move<-initial[j]
      high=0
      stop=c(1,1)
      compare=FALSE
      while (compare==FALSE){
        stop<-append(stop,1)
        s<-length(stop)
        cap1<-numeric()
        for (i in 1:length(bbishop2_move)){
          if (bbishop2_move[i] %in% board2){position<-which(board2==bbishop2_move[i], arr.ind=TRUE)}
          if (bbishop2_move[i] %in% board){position<-which(board==bbishop2_move[i], arr.ind=TRUE)}
          cap1_horse<-bishop(board,position,black,white)}}
      P1_listing<-append(P1_listing,list(mat))}}}
```

```

    cap1<-append(cap1,cap1_horse)
    cap1<-unique(cap1)}
maximum<-length(cap1)
high<-max(high,maximum)
bbishop2_move<-cap1
cap3<-append(cap1,numeric(62-maximum))
if (maximum>0){mat[,s]<-cap3}
if (is.numeric(cap1)==TRUE){break}
if (s>2){compare<-all(mat[,s-2]==cap3)}
if (s==48){break}
mat2<-mat[1:high,]
dm<-dim(mat2)
dn<-dm[1]
if (is.numeric(cap1)==TRUE){dn=1}
group<-append(group,dn)
b2_listing<-append(b2_listing,list(mat2))}}
if ("B_1" %in% board){
p_bishop1<-which(board=="B_1", arr.ind=TRUE)
bbishop1_move<-bishop(board,p_bishop1,black,white)
if (length(changed)>0){
  bbishop1_move<-bbishop1_move[bbishop1_move!=changed]
  bbishop1_move<-bbishop1_move[!is.na(bbishop1_move)]}
initial<-bbishop1_move
if (length(bbishop1_move)>0){
  l=length(bbishop1_move)
  for (j in 1:l){
    mat<-matrix(nrow=62,ncol=49)
    mat[,1]<-c(rep("B_1",62))
    mat[,2]<-c(rep(initial[j],62))
    bbishop1_move<-initial[j]
    high=0
    stop=c(1,1)
    compare=FALSE
    while (compare==FALSE){
      stop<-append(stop,1)
      s<-length(stop)
      cap1<-numeric()
      for (i in 1:length(bbishop1_move)){
        if (bbishop1_move[i] %in% board2){position<-which(board2==bbishop1_move[i], arr.ind=TRUE)}
        if (bbishop1_move[i] %in% board){position<-which(board==bbishop1_move[i], arr.ind=TRUE)}
        cap1_horse<-bishop(board,position,black,white)
        cap1<-append(cap1,cap1_horse)
        cap1<-unique(cap1)}
      maximum<-length(cap1)
      high<-max(high,maximum)
      bbishop1_move<-cap1
      cap3<-append(cap1,numeric(62-maximum))
      if (maximum>0){mat[,s]<-cap3}
      if (is.numeric(cap1)==TRUE){break}
      if (s>2){compare<-all(mat[,s-2]==cap3)}
      if (s==48){break}}
    mat2<-mat[1:high,]
    dm<-dim(mat2)
  }
}

```

```

dn<-dm[1]
if (is.numeric(cap1)==TRUE){dn=1}
group<-append(group,dn)
b1_listing<-append(b1_listing,list(mat2))}}
if ("H_2" %in% board){
p_horse1<-which(board=="H_2", arr.ind=TRUE)
bhorse2_move<-horse(board,p_horse1,black,white)
if (length(changed)>0){
  bhorse2_move<-bhorse2_move[bhorse2_move!=changed]
  bhorse2_move<-bhorse2_move[!is.na(bhorse2_move)]}
initial<-bhorse2_move
if (length(bhorse2_move)>0){
  l=length(bhorse2_move)
  for (j in 1:l){
    mat<-matrix(nrow=62,ncol=49)
    mat[,1]<-c(rep("H_2",62))
    mat[,2]<-c(rep(initial[j],62))
    bhorse2_move<-initial[j]
    high=0
    stop=c(1,1)
    compare=FALSE
    while (compare==FALSE){
      stop<-append(stop,1)
      s<-length(stop)
      cap1<-numeric()
      for (i in 1:length(bhorse2_move)){
        if (bhorse2_move[i] %in% board2){position<-which(board2==bhorse2_move[i], arr.ind=TRUE)}
        if (bhorse2_move[i] %in% board){position<-which(board==bhorse2_move[i], arr.ind=TRUE)}
        cap1_horse<-horse(board,position,black,white)
        cap1<-append(cap1,cap1_horse)
        cap1<-unique(cap1)}
      maximum<-length(cap1)
      high<-max(high,maximum)
      bhorse2_move<-cap1
      cap3<-append(cap1,numERIC(62-maximum))
      if (maximum>0){mat[,s]<-cap3}
      if (is.numeric(cap1)==TRUE){break}
      if (s>2){compare<-all(mat[,s-2]==cap3)}
      if (s==48){break}}
    mat2<-mat[1:high,]
    dm<-dim(mat2)
    dn<-dm[1]
    if (is.numeric(cap1)==TRUE){dn=1}
    group<-append(group,dn)
    h2_listing<-append(h2_listing,list(mat2))}}
if ("H_1" %in% board){
p_horse1<-which(board=="H_1", arr.ind=TRUE)
bhorse1_move<-horse(board,p_horse1,black,white)
if (length(changed)>0){
  bhorse1_move<-bhorse1_move[bhorse1_move!=changed]
  bhorse1_move<-bhorse1_move[!is.na(bhorse1_move)]}
initial<-bhorse1_move
if (length(bhorse1_move)>0){

```

```

l=length(bhorse1_move)
for (j in 1:l){
  mat<-matrix(nrow=62,ncol=49)
  mat[,1]<-c(rep("H_1",62))
  mat[,2]<-c(rep(initial[j],62))
  bhorse1_move<-initial[j]
  high=0
  stop=c(1,1)
  compare=FALSE
  while (compare==FALSE){
    stop<-append(stop,1)
    s<-length(stop)
    cap1<-numeric()
    for (i in 1:length(bhorse1_move)){
      if (bhorse1_move[i] %in% board2){position<-which(board2==bhorse1_move[i], arr.ind=TRUE)}
      if (bhorse1_move[i] %in% board){position<-which(board==bhorse1_move[i], arr.ind=TRUE)}
      cap1_horse<-horse(board,position,black,white)
      cap1<-append(cap1,cap1_horse)
      cap1<-unique(cap1)
    maximum<-length(cap1)
    high<-max(high,maximum)
    bhorse1_move<-cap1
    cap3<-append(cap1,numERIC(62-maximum))
    if (maximum>0){mat[,s]<-cap3}
    if (is.numeric(cap1)==TRUE){break}
    if (s>2){compare<-all(mat[,s-2]==cap3)}
    if (s==48){break}}
    mat2<-mat[1:high,]
    dm<-dim(mat2)
    dn<-dm[1]
    if (is.numeric(cap1)==TRUE){dn=1}
    group<-append(group,dn)
    h1_listing<-append(h1_listing,list(mat2))}}
  if ("R_2" %in% board){
    p_rook1<-which(board=="R_2", arr.ind=TRUE)
    brook2_move<-rook(board,p_rook1,black,white)
    if (length(changed)>0){
      brook2_move<-brook2_move[brook2_move!=changed]
      brook2_move<-brook2_move[!is.na(brook2_move)]}
    initial<-brook2_move
    if (length(brook2_move)>0){
      l=length(brook2_move)
      for (j in 1:l){
        mat<-matrix(nrow=62,ncol=49)
        mat[,1]<-c(rep("R_2",62))
        mat[,2]<-c(rep(initial[j],62))
        brook2_move<-initial[j]
        high=0
        stop=c(1,1)
        compare=FALSE
        while (compare==FALSE){
          stop<-append(stop,1)
          s<-length(stop)

```

```

cap1<-numeric()
for (i in 1:length(brook2_move)){
  if (brook2_move[i] %in% board2){position<-which(board2==brook2_move[i], arr.ind=TRUE)}
  if (brook2_move[i] %in% board){position<-which(board==brook2_move[i], arr.ind=TRUE)}
  cap1_horse<-rook(board,position,black,white)
  cap1<-append(cap1,cap1_horse)
  cap1<-unique(cap1)
  maximum<-length(cap1)
  high<-max(high,maximum)
  brook2_move<-cap1
  cap3<-append(cap1,numeric(62-maximum))
  if (maximum>0){mat[,s]<-cap3}
  if (is.numeric(cap1)==TRUE){break}
  if (s>2){compare<-all(mat[,s-2]==cap3)}
  if (s==48){break}
  mat2<-mat[1:high,]
  dm<-dim(mat2)
  dn<-dm[1]
  if (is.numeric(cap1)==TRUE){dn=1}
  group<-append(group,dn)
  r2_listing<-append(r2_listing,list(mat2))}}
if ("R_1" %in% board){
  p_rook1<-which(board=="R_1", arr.ind=TRUE)
  brook1_move<-rook(board,p_rook1,black,white)
  if (length(changed)>0){
    brook1_move<-brook1_move[brook1_move!=changed]
    brook1_move<-brook1_move[!is.na(brook1_move)]}
  initial<-brook1_move
  if (length(brook1_move)>0){
    l=length(brook1_move)
    for (j in 1:l){
      mat<-matrix(nrow=62,ncol=49)
      mat[,1]<-c(rep("R_1",62))
      mat[,2]<-c(rep(initial[j],62))
      brook1_move<-initial[j]
      high=0
      stop=c(1,1)
      compare=FALSE
      while (compare==FALSE){
        stop<-append(stop,1)
        s<-length(stop)
        cap1<-numeric()
        for (i in 1:length(brook1_move)){
          if (brook1_move[i] %in% board2){position<-which(board2==brook1_move[i], arr.ind=TRUE)}
          if (brook1_move[i] %in% board){position<-which(board==brook1_move[i], arr.ind=TRUE)}
          cap1_horse<-rook(board,position,black,white)
          cap1<-append(cap1,cap1_horse)
          cap1<-unique(cap1)}
        maximum<-length(cap1)
        high<-max(high,maximum)
        brook1_move<-cap1
        cap3<-append(cap1,numeric(62-maximum))
        if (maximum>0){mat[,s]<-cap3}

```

```

    if (is.numeric(cap1)==TRUE){break}
    if (s>2){compare<-all(mat[,s-2]==cap3)}
    if (s==48){break}
  mat2<-mat[1:high,]
  dm<-dim(mat2)
  dn<-dm[1]
  if (is.numeric(cap1)==TRUE){dn=1}
  group<-append(group,dn)
  r1_listing<-append(r1_listing,list(mat2))}}
if ("Q" %in% board){
  p_queen<-which(board=="Q", arr.ind=TRUE)
  bqueen_move<-queen(board,p_queen,black,white)
  if (length(changed)>0){
    bqueen_move<-bqueen_move[bqueen_move!=changed]
    bqueen_move<-bqueen_move[!is.na(bqueen_move)]}
  initial<-bqueen_move
  if (length(bqueen_move)>0){
    l=length(bqueen_move)
    for (j in 1:l){
      mat<-matrix(nrow=62,ncol=49)
      mat[,1]<-c(rep("Q",62))
      mat[,2]<-c(rep(initial[j],62))
      bqueen_move<-initial[j]
      high=0
      stop=c(1,1)
      compare=FALSE
      while (compare==FALSE) {
        stop<-append(stop,1)
        s<-length(stop)
        cap1<-numeric()
        for (i in 1:length(bqueen_move)){
          if (bqueen_move[i] %in% board2){position<-which(board2==bqueen_move[i], arr.ind=TRUE)}
          if (bqueen_move[i] %in% board){position<-which(board==bqueen_move[i], arr.ind=TRUE)}
          cap1_horse<-queen(board,position,black,white)
          cap1<-append(cap1,cap1_horse)
          cap1<-unique(cap1)}
        maximum<-length(cap1)
        high<-max(high,maximum)
        bqueen_move<-cap1
        cap3<-append(cap1,numerical(62-maximum))
        if (maximum>0){mat[,s]<-cap3}
        if (is.numeric(cap1)==TRUE){break}
        if (s>2){compare<-all(mat[,s-2]==cap3)}
        if (s==48){break}
      mat2<-mat[1:high,]
      dm<-dim(mat2)
      dn<-dm[1]
      if (is.numeric(cap1)==TRUE){dn=1}
      group<-append(group,dn)
      q_listing<-append(q_listing,list(mat2))}}
  b_king<-which(board=="K", arr.ind=TRUE)
  bking_move<-king(board,b_king,black,white)
  bking_minus<-Reduce(intersect, list(bking_move,white_move))
}

```

```

if (length(bking_minus)>0){
  for (x in 1:length(bking_minus)){
    separate<-which(bking_move==bking_minus[x], arr.ind=TRUE)
    bking_move<-bking_move[-separate]}
  if (length(changed)>0){
    bking_move<-bking_move[bking_move!=changed]
    bking_move<-bking_move[!is.na(bking_move)]}
initial<-bking_move
if (length(bking_move)>0){
  l=length(bking_move)
  for (j in 1:l){
    mat<-matrix(nrow=62,ncol=49)
    mat[,1]<-c(rep("K",62))
    mat[,2]<-c(rep(initial[j],62))
    bking_move<-initial[j]
    high=0
    stop=c(1,1)
    compare=FALSE
    while (compare==FALSE){
      stop<-append(stop,1)
      s<-length(stop)
      cap1<-numeric()
      for (i in 1:length(bking_move)){
        if (bking_move[i] %in% board2){position<-which(board2==bking_move[i], arr.ind=TRUE)}
        if (bking_move[i] %in% board){position<-which(board==bking_move[i], arr.ind=TRUE)}
        cap1_horse<-king(board,position,black,white)
        cap1<-append(cap1,cap1_horse)
        cap1<-unique(cap1)}
      maximum<-length(cap1)
      high<-max(high,maximum)
      bking_move<-cap1
      cap3<-append(cap1,numERIC(62-maximum))
      if (maximum>0){mat[,s]<-cap3}
      if (is.character(cap1)==TRUE){break}
      if (s>2){compare<-all(mat[,s-2]==cap3)}
      if (s==48){break}}
    mat2<-mat[1:high,]
    dm<-dim(mat2)
    dn<-dm[1]
    if (is.character(cap1)==TRUE){dn=1}
    group<-append(group,dn)
    k_listing<-append(k_listing,list(mat2))}

#combine
total<-list(do.call(rbind,P8_listing),do.call(rbind,P7_listing),do.call(rbind,P6_listing),
            do.call(rbind,P5_listing),do.call(rbind,P4_listing),do.call(rbind,P3_listing),
            do.call(rbind,P2_listing),do.call(rbind,P1_listing),do.call(rbind,b2_listing),
            do.call(rbind,b1_listing),do.call(rbind,h2_listing),do.call(rbind,h1_listing),
            do.call(rbind,r2_listing),do.call(rbind,r1_listing),do.call(rbind,q_listing),
            do.call(rbind,k_listing))
grand<-do.call(rbind,total)
great<-grand[,colSums(is.na(grand))<nrow(grand)]
lg<-length(group)

```

```

scores<-great[,1:2]
score_black<-scores[!duplicated(scores),]
scoring_black<-numeric(lg)

#defense
if ("K" %in% score_white[,2]){
  check<-which(score_white[,2]=="K", arr.ind=TRUE)
  score_black1<-score_black[score_black[,1]=="K",]
  if (length(check)==1){
    score1<-score_black[score_black[,2]==score_white[check,1],]
    score_black2<-do.call(rbind,list(score_black1,score1))
    if (score_white[check,1] %in% more_white){
      checking<-score_white[check,1]
      start<-which(board==checking, arr.ind=TRUE)
      end<-which(board=="K", arr.ind=TRUE)
      if (start[1]>end[1] && start[2]>end[2]){
        shape<-board2[end[1]:start[1],end[2]:start[2]]
        shape2<-diag(shape)
        if (end[1]>1 && end[2]>1){
          if (board[end[1]-1,end[2]-1]== ""){extra<-board2[end[1]-1,end[2]-1]}
          if (board[end[1]-1,end[2]-1] %in% black){extra<-extra}
          if (board[end[1]-1,end[2]-1] %in% white){extra<-board2[end[1]-1,end[2]-1]}}
        if (start[1]>end[1] && start[2]<end[2]){
          shape<-board2[end[1]:start[1],start[2]:end[2]]
          shape2<-diag(shape)
          if (end[1]>1 && end[2]<8){
            if (board[end[1]-1,end[2]+1]== ""){extra<-board2[end[1]-1,end[2]+1]}
            if (board[end[1]-1,end[2]+1] %in% black){extra<-extra}
            if (board[end[1]-1,end[2]+1] %in% white){extra<-board2[end[1]-1,end[2]+1]}}
        if (start[1]<end[1] && start[2]>end[2]){
          shape<-board2[start[1]:end[1],end[2]:start[2]]
          shape2<-diag(shape)
          if (end[1]<8 && end[2]>1{
            if (board[end[1]+1,end[2]-1]== ""){extra<-board2[end[1]+1,end[2]-1]}
            if (board[end[1]+1,end[2]-1] %in% black){extra<-extra}
            if (board[end[1]+1,end[2]-1] %in% white){extra<-board2[end[1]+1,end[2]-1]}}
        if (start[1]<end[1] && start[2]<end[2]){
          shape<-board2[start[1]:end[1],start[2]:end[2]]
          shape2<-diag(shape)
          if (end[1]<8 && end[2]<8){
            if (board[end[1]+1,end[2]+1]== ""){extra<-board2[end[1]+1,end[2]+1]}
            if (board[end[1]+1,end[2]+1] %in% black){extra<-extra}
            if (board[end[1]+1,end[2]+1] %in% white){extra<-board2[end[1]+1,end[2]+1]}}
        if (start[1]==end[1] && start[2]>end[2]){
          shape2<-board2[start[1],end[2]:start[2]]
          if (end[1]>1 && end[2]>2{
            if (board[end[1]-1,end[2]-1]== ""){extra<-board2[end[1]-1,end[2]-1]}
            if (board[end[1]-1,end[2]-1] %in% black){extra<-extra}
            if (board[end[1]-1,end[2]-1] %in% white){extra<-board2[end[1]-1,end[2]-1]}}
        if (start[1]==end[1] && start[2]<end[2]){
          shape2<-board2[start[1],start[2]:end[2]]
          if (end[1]>1 && end[2]>2){

```

```

    if (board[end[1]-1,end[2]-1]== ""){extra<-board2[end[1]-1,end[2]-1]}
    if (board[end[1]-1,end[2]-1] %in% black){extra<-extra}
    if (board[end[1]-1,end[2]-1] %in% white){extra<-board2[end[1]-1,end[2]-1]}}
if (start[1]<end[1] && start[2]==end[2]){
  shape2<-board2[start[1]:end[1],start[2]]
  if (end[1]>1 && end[2]>2){
    if (board[end[1]-1,end[2]-1]== ""){extra<-board2[end[1]-1,end[2]-1]}
    if (board[end[1]-1,end[2]-1] %in% black){extra<-extra}
    if (board[end[1]-1,end[2]-1] %in% white){extra<-board2[end[1]-1,end[2]-1]}}
  if (start[1]>end[1] && start[2]==end[2]){
    shape2<-board2[end[1]:start[1],start[2]]
    if (end[1]>1 && end[2]>2){
      if (board[end[1]-1,end[2]-1]== ""){extra<-board2[end[1]-1,end[2]-1]}
      if (board[end[1]-1,end[2]-1] %in% black){extra<-extra}
      if (board[end[1]-1,end[2]-1] %in% white){extra<-board2[end[1]-1,end[2]-1]}}}
  shape3<-shape2[-length(shape2)]
  shape3<-shape3[-1]
  if (length(shape3)>0){
    for (y in 1:length(shape3)){
      score2<-score_black[score_black[,2]==shape3[y],]
      score_black2<-do.call(rbind, list(score_black2,score2))}}
  score_black<-score_black2
  if (length(score_black)==0){respond="Checkmate! You Win!"}
  if (length(check)>1){
    if (length(score_black1)==0){
      respond="Checkmate! You Win!"
      score_black<-numeric()
      if (length(score_black1)!=0){score_black=score_black1}}
    lg2<-dim(score_black)
    lg<-lg2[1]}
  if (length(score_black)!=0){

#scoring
scoring_black<-numeric(lg)
for (i in 1:lg){
  if (score_black[i,2] %in% board2){
    if (score_black[i,2] %in% score_white[,2] == FALSE){scoring_black[i]=0}
    if (score_black[i,2] %in% score_white[,2]){
      pose<-which(black %in% score_black[i,1])
      scoring_black[i]=-1*b_score[pose]}}
  if (score_black[i,2] %in% board){
    if (score_black[i,2] %in% score_white[,2] == FALSE){
      pose2<-which(white %in% score_black[i,2])
      scoring_black[i]=w_score[pose2]}
    if (score_black[i,2] %in% score_white[,2]){
      pose3<-which(white %in% score_black[i,2])
      pose4<-which(black %in% score_black[i,1])
      scoring_black[i]=w_score[pose3]-b_score[pose4]}}}
common<-Reduce(intersect, list(black,score_white[,2]))
if (length(common)>0){
  basis<-numeric()
  for (i in 1:length(common)){
    base<-b_score[which(black %in% common[i])]
```

```

basis<-append(basis,base)
base_score<-max(basis,na.rm=TRUE)}
pattern<-which(basis %in% base_score)
subtract<-numeric()
for (j in 1:length(pattern)){
  subtract<-append(subtract,which(score_black[,1] %in% common[[pattern[j]]]))}
for (k in 1:length(subtract)){
  if (base_score==1000){base_score=2000}
  scoring_black[subtract[k]]=scoring_black[subtract[k]]+base_score}

#checkmate
g=0
respond<-numeric()
mate<-numeric()
for (j in 1:lg){
  primary<-great[(g+1):(g+group[j]),3]
  prime<-numeric()
  for (k in 1:length(target)){
    if (target[k] %in% primary){prime<-append(prime,target[k])}}
  if (length(prime)>0){
    if (all(target==prime)==TRUE){
      if (scoring_black[j]>=0){mate<-append(mate,j)}}
  g<-g+group[j]}
if (length(mate)>0){
  response<-which(board==score_black[mate[1],1], arr.ind=TRUE)
  if (score_black[mate[1],2] %in% board){
    r<-which(board==score_black[mate[1],2], arr.ind=TRUE)
    score_black[mate[1],2]=board2[r[1],r[2]]}
  respond<-c(board2[[response[1],response[2]],score_black[mate[1],2]])}

#analysis
if (length(mate)==0){
  greater<-great[,-c(1:2)]
  dg<-dim(greater)
  target3<-target
  for (i in 1:dg[2]){
    for (j in 1:dg[1]){
      if (greater[j,i] %in% target3){
        spot<-which(target3 %in% greater[j,i])
        target3<-target3[-spot]}}
    if (length(target3)==0){break}}
  greatest<-great[,1:(i+2)]}
g=0
mate1<-numeric()
mate2<-numeric()
for (j in 1:lg){
  primary<-greatest[(g+1):(g+group[j]),]
  prime1=0
  prime2=0
  if (length(target2)>0){
    for (k in 1:length(target2)){
      if (target2[k] %in% primary){prime1=1}}}
  if (prime1==1){mate1<-append(mate1,j)}}
```

```

    if ("k" %in% primary){prime2=1}
    if (prime2==1){mate2<-append(mate2,j)}
    g<-g+group[j]
mate3<-append(mate1,mate2)
mate3<-unique(mate3)
sb<-unique(scoring_black)
m3<-numeric()
m4<-numeric()
for (a in 1:length(sb)){
  m<-which.max(scoring_black)
  m2<-which(scoring_black %in% scoring_black[m])
  m3<-append(m3,m2)
  m4<-append(m4,length(m2))
  for (i in 1:length(m2)){
    scoring_black[m2[i]]=NA}

#decision
if (length(mate3)>0){
  h=0
  for (i in 1:length(m4)){
    plus<-m3[(h+1):(h+m4[i])]
    for (j in 1:length(plus)){
      if (plus[j] %in% mate3){
        response<-which(board==score_black[plus[j],1], arr.ind=TRUE)
        if (score_black[plus[j],2] %in% board){
          r<-which(board==score_black[plus[j],2], arr.ind=TRUE)
          score_black[plus[j],2]=board2[r[1],r[2]]}
        respond<-c(board2[response[1],response[2]],score_black[plus[j],2])
        break}}
      if (length(respond)>0){break}
      if (length(respond)==0){
        response<-which(board==score_black[plus[1],1], arr.ind=TRUE)
        if (score_black[plus[1],2] %in% board){
          r<-which(board==score_black[plus[1],2], arr.ind=TRUE)
          score_black[plus[1],2]=board2[r[1],r[2]]}
        respond<-c(board2[response[1],response[2]],score_black[plus[1],2])
        break}
      h<-h+m4[i]}}
  if (length(mate3)==0){
    response<-which(board==score_black[m3[1],1], arr.ind=TRUE)
    if (score_black[m3[1],2] %in% board){
      r<-which(board==score_black[m3[1],2], arr.ind=TRUE)
      score_black[m3[1],2]=board2[r[1],r[2]]}
    respond<-c(board2[response[1],response[2]],score_black[m3[1],2])}}
return(respond)}

#select play as black

#play game as white
from="D1"
to="D6"
BM<-black_move(from,to,board)
board <- move_piece(board,from,to)
board <- move_piece(board,BM[1],BM[2])

```

```
BM  
## [1] "E7" "D6"  
#play game as black
```