```
jokenpo <- function(n.partidas = 1, estilo = 1) # Função com argumentos de
número de partidas e estilo de partida
    {                                                                   #
Abre a função
# 1) Verificação de parâmentros:                                       # A
função deve parar se:
  if(n.partidas <= 0)                                                   # 1)
"n.partidas" for menor ou igual a zero
     stop("Número de partidas deve ser maior e diferente de zero!")   # Para
a função e retorna a frase
  if(class(n.partidas) != "numeric" )                                   # 2) a
classe de "n.partidas" for diferente de "numeric"
     stop("Número de partidas deve ser numérico")                      # Para
a função e retorna a frase
  y = as.integer(n.partidas)                                            # 3)
Quando transformado o "n.partidas" em "integer"
     if(n.partidas != y)                                                #  e
seu número for alterado
        stop("O número de partidas precisa ser inteiro")               # Para
a função e retorna a frase
  if(estilo < 1 | estilo > 2 | class(estilo) != "numeric" )             # 4)
Se "estilo" for diferente de 1 e 2 e da classe numérico
     stop("O estilo da partida deve ser 1 para o jogo
     tradicional ou 2 para o jogo mais complexo")                      # Para
a função e retorna a frase
  w = as.integer(estilo)                                                # 5)
Quando transformado o "estilo" em "integer"
     if(estilo != w)                                                    #  e
seu número for alterado
        stop("O número do estilo precisa ser inteiro")                 # Para
a função e retorna a frase

# 2) Preparação de tabelas de pontos e de partidas:

# 2.1) Tabela de pontos do operador e do computador:
    tab.pontos <- matrix( 0, nrow = 1, ncol = 2,                  # Matriz de
uma linha e duas colunas preenchidas por zeros no objeto "tab.pontos"
      dimnames = list("Pontos",c("Operador","Computador")))    # Linha
denominado de "pontos" e colunas como "Operador" e "Computador"
# 2.3) Criando uma tabela para os resultados do ciclo:
  partidas <- matrix(NA, nrow = n.partidas, ncol = 3,          # Matriz de
"n.partidas" linhas e três colunas preenchidas por NAs
      dimnames = (list((1:n.partidas),                         # Linhas
denomidadas por números de 1 a "n.partidas"
      c("Operador","Computador","Resultado"))))                # Colunas
denominadas por "Operador","Computador" e "Resultado"

# 3) Carregamento do pacote de imagens e cores:
verifica <- "magick" %in% rownames(installed.packages())       # Verificando
se o pacote "magick" está instalado e salvando no objeto "verifica"
if (verifica == FALSE)                                         # Se
```

```
"verifica" for "False":
    {install.packages("magick")}                              # Instala o
pacote "magick"
library(magick)                                               # Acessa o
pacote "magick"
library(crayon)                                               # Acessa o
pacote "crayon"

# 4) Lista de imagens a serem executadas:
bem.vindo <- image_read(
# Salva a imagem do link no objeto "bem.vindo"
"https://www.danny.com.br/wp-content/uploads/2015/12/imagem-branca-grande.pn
g")%>% # Busca a imagem em branco da internet e reproduz
 image_annotate("           Bem vindo ao Jokenpo! ",
# Escreve uma mensagem de boas vindas em cima da imagem branca,
  size = 30, color = "DeepSkyBlue4", font = "Comic Sans")
# de tamanho 30, da cor "DeepSkyBlue4", em fonte comic sans

  if(estilo == 1)
# Se o "estilo" é igual a 1
  {
  bem.vindo1 <- image_annotate(bem.vindo,"\n\n   Você escolheu a partida
tradicional!\n   Escolha entre pedra, papel e tesoura mas saiba que:\n
1) Pedra entorta tesoura,\n     2) Tesoura corta papel e\n     3) Papel
cobre pedra", # Inclui as frases baseado em "bem.vindo"
  size = 20, color = "MediumSeaGreen", font = "Comic Sans")
# em tamanho 20, cor "MediumSeaGreen",e fonte comic sans
  }

  if(estilo == 2)                                                 # Se
"estilo" for igual a 2
  {
bem.vindo2 <- image_annotate(bem.vindo,"\n\n   Você escolheu a partida
complexa!\n   Escolha entre pedra, papel, tesoura, spock e lagarto\n  mas
saiba que:\n    1) Pedra entorta tesoura       6) Spock quebra tesoura\n
2) Tesoura corta papel          7) Tesoura decapta lagarto\n    3) Papel
cobre pedra             8) Lagarto come papel\n    4) Pedra esmaga lagarto
9) Papel contesta Spock\n    5) Lagarto envenena Spock   10) Spock vaporiza
pedra",                                          # Inclui as frases baseado
em "bem.vindo" e salva no objeto "bem.vindo2"
size = 20, color ="MediumSeaGreen", font = "Comic Sans")            # em
tamanho 20, cor "MediumSeaGreen",e fonte comic sans
  }
derrota <- image_read(
# Salva a imagem no objeto "derrota"
 "https://segredosdomundo.r7.com/wp-content/uploads/2015/12/5-11.jpg") %>%
# Busca a imagem na internet
 image_scale("350") %>%
# Modifica o tamanho da imagem para 350
 image_annotate("        Que pena, não foi dessa vez! \n            Mais
```

```
sorte na próxima! ", # Inlcui a frase na imagem
 size = 20, color = "blue", font="Comic Sans")
# Em tamanho 20, cor azul, e fonte comic sans

derrota.2 <- image_read(
# Salva a imagem no objeto "derrota.2"
"http://www.braian.com.br/wp-content/uploads/2012/01/pedraxtesoura1.jpg")
%>%  # Busca a imagem na internet
 image_scale("350") %>%
# Modifica o tamanho da imagem para 350
 image_annotate("   Keep Calm    \n and Carry on! ",
# Inclui a frase na imagem
  size = 20, color = "white", font="Comic Sans",
# Em tamanho 20, cor branco, e fonte comic sans
  boxcolor = "black", location = "+120+250")
# Em caixa de texto com fundo preto no local 120 por 250

vitoria <- image_read(
# Salva no objeto "vitoria"
 "https://cdn-images-1.medium.com/max/384/1*1rMVnfCZPbJh_AupNyem-w.jpeg")
%>% # Busca a imagem na internet
image_annotate("Parabéns! ", size = 30, color = "red", font="Comic Sans",
# Anota a frase em tamanho 30, cor vermelha e fonte comic sans
 boxcolor = "pink", location = "+70+290")
# com fundo da caixa de texto em pink e localização 70 por 290

vitoria.2 <- image_read(
# Salva no objeto "vitoria.2"
"https://ih1.redbubble.net/image.482463497.9377/flat,750x,075,f-pad,750x1000
,f8f8f8.jpg")  %>% # Busca a imagem na internet
image_scale("350") %>%
# Modifica o tamanho para 350
image_annotate("GG (Good Game!)", size = 40, color = "white",
# Inclui a frase no tamanho 40 e em branco
 font="Comic Sans", location = "+30+290")
# fonte comic sans na localização 30 por 290

##########################################################################
#######################################################
# 5) Partida para "estilo 1 e 2":

if(estilo == 1 | estilo == 2)                                        # Para
"estilo" igual a 1 e 2
    {                                                                # É
criado uma tabela de possíveis resultados
    if(estilo == 1)                                                  # Se
"estilo" é igual a 1
        {
        vet <- c("empate","ganha","perde","perde","empate",         #
Primeiro: organiza as informações da tabela de resultados em sequência
              "ganha","ganha","perde","empate")
```

```
        linhas <- c("pedra","papel","tesoura")                       # As
linhas da matriz referem-se às opções do operador
        colunas <- c("pedra","papel","tesoura")                      # As
colunas da matriz referem-se às opções do computador
        tab.resul <- matrix( data = vet, nrow = 3, ncol = 3,         # Por
fim, é gerada uma matriz com 3 linhas por 3 colunas,
                  byrow = FALSE,                                     #
                  dimnames = list(linhas,colunas))                   #
denominadas pelos objetos "linhas" e "colunas"
        print(bem.vindo1)                                            #
Reproduz a imagem criada em "bem.vindo1"
        }
    if(estilo == 2)                                                 # Se o
"estilo" é igual a 2
        {
         vet <- c("empate","ganha","perde","ganha","perde",          #
Primeiro: organiza as informações da tabela de resultados em sequência
             "perde","empate","ganha", "perde","ganha",
             "ganha","perde","empate","ganha","perde",
             "perde","ganha","perde","empate","ganha",
             "ganha","perde","ganha","perde","empate")
        linhas <- c("pedra","papel","tesoura","spock","lagarto")         #
As linhas da matriz referem-se às opções do operador
        colunas <- c("pedra","papel","tesoura","spock","lagarto")        #
As colunas da matriz referem-se às opções do computador
        tab.resul <- matrix( data = vet, nrow = 5, ncol = 5,             #
Por fim, é gerada uma matriz com 3 linhas por 3 colunas,
                       byrow = FALSE,                                    #
preenchida por ordem de colunas
                           dimnames = list(linhas,colunas))
#    denominadas pelos objetos "linhas" e "colunas"
        print(bem.vindo2)                                                #
Reproduz a imagem guardada em "bem.vindo2"
        }
     for(i in 1:n.partidas)                                             #
Criando o ciclo for() que vai do número 1 até "n.partidas"
     {
     if(interactive())
# Com o comando interativo verdadeiro
         {
         if(estilo == 1)
# Para o "estilo" igual a 1
             {
             arma.op <- readline("ESCOLHA SUA ARMA: -- pedra/papel/tesoura:
")   # Salva a resposta interativa no objeto "arma.op"
             if(arma.op == "pedra" | arma.op == "papel" | arma.op ==
"tesoura")  # Se a digitação for correta
                {partidas[i,1] <- arma.op                          #
Transfere o "arma.op" para 1ª coluna, linha "i" de "partidas"
         } else {
```

```
# Ou
             repeat
# Repete a sequencia de comandos abaixo:
                 {
                 cat(red$bold("VOCÊ DIGITOU ALGO ERRADO, DIGITE NOVAMENTE:\n
")) # Mostra a frase em vermelho e negrito
                 if(interactive())
# Repete o comando interativo
                     {
                     arma.op <- readline("ESCOLHA SUA ARMA --
pedra/papel/tesoura: ")  # Salva a resposta interativa no objeto "arma.op"
                     if(arma.op == "pedra" | arma.op == "papel" | arma.op ==
"tesoura")# Se a digitação for correta
                         {partidas[i,1] <- arma.op
# Transfere o "arma.op" para 1ª coluna,linha "i" de "partidas"
                          break()}}}}
# Finaliza o ciclo de repetição quando "arma.op" estiver correta
                                                                          #
Sorteio do computador:
        arma.comp <- sample(c("pedra","papel", "tesoura"), 1)         #
Sorteia aleatóriamente por 1 vez e salva no objeto "arma.comp"
            }
        if(estilo == 2)
# Para o "estilo" igual a 2
            {
            arma.op <- readline("ESCOLHA SUA ARMA: --
pedra/papel/tesoura/spock/lagarto: ") # Salva a resposta interativa no
objeto "arma.op"
            if(arma.op == "pedra" | arma.op == "papel" | arma.op ==
"tesoura" |             # Se a digitação for correta
                arma.op == "spock" | arma.op == "lagarto" )
                {partidas[i,1] <- arma.op
# Transfere o "arma.op" para 1ª coluna, linha "i" de "partidas"
            } else {
# Ou
                repeat
# Repete a sequencia de comandos abaixo:
                {
                cat(red$bold("VOCÊ DIGITOU ALGO ERRADO, DIGITE NOVAMENTE:\n
")) # Mostra a frase em vermelho e negrito
                if(interactive())
# Repete o comando interativo
                      {
                      arma.op <- readline("ESCOLHA SUA ARMA --
pedra/papel/tesoura/spock/lagarto: ")# Salva a resposta interativa no objeto
"arma.op"
                      if(arma.op == "pedra" | arma.op == "papel" | arma.op
== "tesoura" |
                                 arma.op == "spock" | arma.op == "lagarto")
# Se a digitação for correta
                          {partidas[i,1] <- arma.op            # Transfere
```

```
o "arma.op" para 1ª coluna, linha "i" de "partidas"
                      break()}}}}                                # Finaliza
o ciclo de repetição quando "arma.op" estiver correta
                                                                 # Sorteio
do computador:
    arma.comp <- sample(c("pedra","papel", "tesoura","spock","lagarto"), 1)
# Sorteia aleatóriamente por 1 vez e salva no objeto "arma.comp"
          }
    partidas[i,2] <- arma.comp                                   # Transfere
"arma.comp" para a 2ª coluna, linha "i" de "partidas"
                                                                 #
Comparando resultados:
    resultado <- tab.resul[arma.op,arma.comp]                    # Procura
os dois objetos na matriz "tab.resul"
              }
# Se o resultado for "empate", repete-se a operação da escolha interativa,
do sorteio do computador e da procura do resultado:
    if(resultado == "empate")                                    # Se
"resultado" for igual a "empate"
        {
        repeat                                                   # Repete os
comandos
        {
            cat(green$bold("DEU EMPATE! REPITA A JOGADA E ESCOLHA
NOVAMENTE:\n ")) # Reproduz a mensagem em verde e negrito
            if(interactive())                                    # Repete o
comando interativo
                    {
                    if(estilo == 1)                              # Se
"estilo" é igual a 1
                    {
                    arma.op <- readline("ESCOLHA SUA ARMA --
pedra/papel/tesoura: ") # Salva a resposta interativa no objeto "arma.op"
                    if(arma.op == "pedra" | arma.op == "papel" | arma.op ==
"tesoura") # Se a digitação for correta
                        {partidas[i,1] <- arma.op                # Transfere
o "arma.op" para 1ª coluna, linha "i" de "partidas"
                    } else {                                     # Ou
                        repeat{                                  # Repete os
comandos
                         cat(red$bold("VOCÊ DIGITOU ALGO ERRADO, DIGITE
NOVAMENTE:\n ")) # Mostra a frase em vermelho e negrito
                        if(interactive())                        # Repete o
comando interativo
                              {
                              arma.op <- readline("ESCOLHA SUA ARMA --
pedra/papel/tesoura: ") # Salva a resposta interativa no objeto "arma.op"
                              if(arma.op == "pedra" | arma.op == "papel" |
arma.op == "tesoura") # Se a digitação for correta
                                 {partidas[i,1] <- arma.op                #
```

```
Transfere o "arma.op" para 1ª coluna, linha "i" de "partidas"
                                  break()}}}}                                #
Finaliza o ciclo de repetição quando "arma.op" estiver correta
                    arma.comp <- sample(c("pedra","papel", "tesoura"), 1)
# Sorteia aleatóriamente por 1 vez e salva no objeto "arma.comp"
                    partidas[i,2] <- arma.comp                               #
Transfere "arma.comp" para a 2ª coluna, linha "i" de "partidas"
                              resultado <- tab.resul[arma.op,arma.comp]
# Procura os dois objetos na matriz "tab.resul"
                          if(resultado != "empate") break()
# Se "resultado" é diferente de "empate", quebra o ciclo de repetição
                    }
                    if(estilo == 2)                                          #
Se "estilo" é igual a 2
                    {
                        arma.op <- readline("ESCOLHA SUA ARMA: --
pedra/papel/tesoura/spock/lagarto: ")# Salva a resposta interativa no objeto
"arma.op"
                        if(arma.op == "pedra" | arma.op == "papel" | arma.op
== "tesoura" | # Se a digitação for correta
                            arma.op == "spock" | arma.op == "lagarto" )
                        {partidas[i,1] <- arma.op
# Transfere o "arma.op" para 1ª coluna, linha "i" de "partidas"
                        } else {
# Ou
                        repeat
# Repete os comandos
                         {
                         cat(red$bold("VOCÊ DIGITOU ALGO ERRADO, DIGITE
NOVAMENTE:\n ")) # Mostra a frase em vermelho e negrito
                          if(interactive())
# Repete o comando interativo
                          {
                          arma.op <- readline("ESCOLHA SUA ARMA --
pedra/papel/tesoura/spock/lagarto: ")# Salva a resposta interativa no objeto
"arma.op"
                          if(arma.op == "pedra" | arma.op == "papel" |
arma.op == "tesoura" |           # Se a digitação for correta
                                 arma.op == "spock" | arma.op == "lagarto")
                           {partidas[i,1] <- arma.op                         #
Transfere o "arma.op" para 1ª coluna, linha "i" de "partidas"
                            break()}}}}                                      #
Finaliza o ciclo de repetição quando "arma.op" estiver correta
                    arma.comp <- sample(c("pedra","papel", "tesoura",
"spock","lagarto"), 1)# Sorteia aleatóriamente por 1 vez e salva no objeto
"arma.comp"
                    partidas[i,2] <- arma.comp                               #
Transfere "arma.comp" para a 2ª coluna, linha "i" de "partidas"
                              resultado <- tab.resul[arma.op,arma.comp]
# Procura os dois objetos na matriz "tab.resul"
                          if(resultado != "empate") break()}}}}
```

```
# Se "resultado" é diferente de "empate", quebra o ciclo de repetição

# Se resultado for "ganha":
if(resultado == "ganha")                                          # Se
"resultado" é igual a "ganha"
    {
    partidas[i,3] <- resultado                                    # Guarda a
palavra na 3ª coluna, linha "i" de "partidas"
    cat(cyan$bold("\nPARABÉNS! VOCÊ É O VENCEDOR!\n "))         # Mostra a
frase em cor "cyan" e negrito
    tab.pontos[1,1] <- (tab.pontos[1,1]+1)                        # Acrescenta
1 ponto na 1ª coluna de "tab.pontos"
    print(sample(c(vitoria, vitoria.2),1))                        # Reproduz
aleatoriamente uma das imagens salvas em "vitoria" e "vitoria.2"
    }
# Se o resultado for perde:
if(resultado == "perde")
# Se "resultado" é igual a "perde"
    {
    partidas[i,3] <- resultado
# Guarda a palavra na 3ª coluna, linha "i" de "partidas"
    cat(magenta$bold("ESSA FOI POR POUCO... MAIS SORTE DA PRÓXIMA VEZ\n")) #
Mostra a frase em cor magenta e negrito
    tab.pontos[1,2] <- (tab.pontos[1,2]+1)
# Acrescenta 1 ponto na 2ª coluna de "tab.pontos"
    print(sample(c(derrota, derrota.2),1))                            #
Reproduz aleatoriamente uma das imagens salvas em "derrota" e "derrota.2"
    }

# Quando a PONTUAÇÃO de "tab.pontos" for empatada e for o último jogo:
if(i == n.partidas & tab.pontos[1,2] == tab.pontos[1,1])
# Se a PONTUAÇÃO de "tab.pontos" for empatada e for o último jogo
    {
    print(tab.pontos)
# Mostra a "tab.pontos"
    print(partidas)
# Mostra a "partidas"
    cat(green$bold("OS PONTOS FICARAM EMPATADOS, UMA PARTIDA DE DESEMPATE
FOI ACRESCENTADA: ")) # Mostra a frase em gerde e netrito
    desempate <- matrix(NA, nrow = 1, ncol = 3)
# Cria uma nova matriz com 1 linha e 3 colunas
    partidas <- rbind(partidas, desempate)
# Acrescenta esta nova linha em "partidas"
    if(interactive())
# Repete-se todos os comandos acima para apenas uma partida
        {
        if(estilo == 1)
# Se "estilo" é igual a 1
        {
        arma.op <- readline("ESCOLHA SUA ARMA  -- pedra/papel/tesoura: ")
```

```
# Salva a resposta interativa no objeto "arma.op"
        if(arma.op == "pedra" | arma.op == "papel" | arma.op == "tesoura")
# Se a digitação for correta
            {partidas[(n.partidas+1),1] <- arma.op                  #
Transfere o "arma.op" para 1ª coluna, linha "n.partidas+1" de "partidas"
        } else
# Ou
            {
            repeat
# Repete os comandos
                {
                cat(red$bold("VOCÊ DIGITOU ALGO ERRADO, DIGITE NOVAMENTE:\n
")) # Mostra a frase em vermelho e negrito
                if(interactive())
# Repete o comando interativo
                     {
                     arma.op <- readline("ESCOLHA SUA ARMA --
pedra/papel/tesoura: ") # Salva a resposta interativa no objeto "arma.op"
                     if(arma.op == "pedra" | arma.op == "papel" | arma.op ==
"tesoura") # Se a digitação for correta
                         {partidas[(n.partidas+1),1] <- arma.op      #
Transfere o "arma.op" para 1ª coluna, linha "n.partidas+1" de "partidas"
                         break()}                                    #
Finaliza o ciclo de repetição quando "arma.op" estiver correta
               }}}
        arma.comp <- sample(c("pedra","papel", "tesoura"), 1)         #
Sorteia aleatóriamente por 1 vez e salva no objeto "arma.comp"
        }
        if(estilo == 2)                                               # Se
"estilo" é igual a 2
        {
        arma.op <- readline("ESCOLHA SUA ARMA  --
pedra/papel/tesoura/spock/lagarto: ") # Salva a resposta interativa no
objeto "arma.op"
        if(arma.op == "pedra" | arma.op == "papel" | arma.op == "tesoura"
# Se a digitação for correta
                    | arma.op == "spock" | arma.op == "lagarto")
            {partidas[(n.partidas+1),1] <- arma.op                  #
Transfere o "arma.op" para 1ª coluna, linha "n.partidas+1" de "partidas"
        } else                                                        # Ou
            {
            repeat                                                    # Repete
os comandos
                {
                cat(red$bold("VOCÊ DIGITOU ALGO ERRADO, DIGITE NOVAMENTE:\n
"))# Mostra a frase em vermelho e negrito
                if(interactive())
# Repete o comando interativo
                     {
                     arma.op <- readline("ESCOLHA SUA ARMA --
pedra/papel/tesoura/spock/lagarto: ") # Salva a resposta interativa no
```

```
objeto "arma.op
                      if(arma.op == "pedra" | arma.op == "papel" | arma.op ==
"tesoura"   # Se a digitação for correta
                                 | arma.op == "spock" | arma.op == "lagarto")
                          {partidas[(n.partidas+1),1] <- arma.op    #
Transfere o "arma.op" para 1ª coluna, linha "n.partidas+1" de "partidas"
                          break()}                                  #
Finaliza o ciclo de repetição quando "arma.op" estiver correta
               }}}
# Sorteio do computador:
    arma.comp <- sample(c("pedra","papel", "tesoura","spock","lagarto"), 1)
# Sorteia aleatóriamente por 1 vez e salva no objeto "arma.comp"
           }
    partidas[(n.partidas+1),2] <- arma.comp
# Transfere "arma.comp" para a 2ª coluna, linha "i" de "partidas"
                                                                      #
Comparando resultados:
    resultado <- tab.resul[arma.op,arma.comp]                         # Procura
os dois objetos na matriz "tab.resul"
           }
# Se o resultado for empate, repete-se a operação da escolha interativa, o
sorteio do computador e a procura do resultado:
if(resultado == "empate")
# Se "resultado" for igual a "empate"
    {
    repeat
# Repete os comandos
    {
        cat(green$bold("DEU EMPATE! REPITA A JOGADA E ESCOLHA NOVAMENTE: "))
# Reproduz a mensagem em verde e negrito
        if(interactive())
# Repete o comando interativo
                {
                if(estilo == 1)
# Se "estilo" é igual a 1
                {
                arma.op <- readline("ESCOLHA SUA ARMA --
pedra/papel/tesoura: ")    # Salva a resposta interativa no objeto "arma.op
                if(arma.op == "pedra" | arma.op == "papel" | arma.op ==
"tesoura")  # Se a digitação for correta
                    {partidas[(n.partidas+1),1] <- arma.op
# Transfere o "arma.op" para 1ª coluna, linha "n.partidas+1" de "partidas"
                } else
# Ou
                   {
                    repeat{
# Repete os comandos
                    cat(red$bold("VOCÊ DIGITOU ALGO ERRADO, DIGITE
NOVAMENTE:\n "))# Mostra a frase em vermelho e negrito
                    if(interactive())
```

```
# Repete o comando interativo
                          {
                          arma.op <- readline("ESCOLHA SUA ARMA --
pedra/papel/tesoura: ")   # Salva a resposta interativa no objeto "arma.op
                          if(arma.op == "pedra" | arma.op == "papel" |
arma.op == "tesoura") # Se a digitação for correta
                             {partidas[(n.partidas+1),1] <- arma.op
# Transfere o "arma.op" para 1ª coluna, linha "n.partidas+1"
                             break()}
# Finaliza o ciclo de repetição quando "arma.op" estiver correta
                          }}}
                arma.comp <- sample(c("pedra","papel", "tesoura"), 1)
# Sorteia aleatóriamente por 1 vez e salva no objeto "arma.comp"
                partidas[(n.partidas+1),2] <- arma.comp
# Transfere "arma.comp" para a 2ª coluna, linha "i" de "partidas"
                         resultado <- tab.resul[arma.op,arma.comp]
# Procura os dois objetos na matriz "tab.resul"
                      if(resultado != "empate") break()
# Se "resultado" é diferente de "empate", quebra o ciclo de repetição
                }
                if(estilo == 2)                                          #
Se "estilo" é igual a 2
                {
                arma.op <- readline("ESCOLHA SUA ARMA --
pedra/papel/tesoura/spock/lagarto: ") # Salva a resposta interativa no
objeto "arma.op
                if(arma.op == "pedra" | arma.op == "papel" | arma.op ==
"tesoura" |            # Se a digitação for correta
                           arma.op == "spock" | arma.op == "lagarto")
                    {partidas[(n.partidas+1),1] <- arma.op
# Transfere o "arma.op" para 1ª coluna, linha "n.partidas+1"
                } else
                   {
                    repeat{
                    cat(red$bold("VOCÊ DIGITOU ALGO ERRADO, DIGITE
NOVAMENTE:\n "))# Mostra a frase em vermelho e negrito
                    if(interactive())
# Repete o comando interativo
                          {
                          arma.op <- readline("ESCOLHA SUA ARMA --
pedra/papel/tesoura/spock/lagarto: ") # Salva a resposta interativa no
objeto "arma.op
                          if(arma.op == "pedra" | arma.op == "papel" |
arma.op == "tesoura" | # Se a digitação for correta
                                       arma.op == "spock" | arma.op ==
"lagarto")
                                    {partidas[(n.partidas+1),1] <- arma.op   #
Transfere o "arma.op" para 1ª coluna, linha "n.partidas+1"
                             break()}                                    #
Finaliza o ciclo de repetição quando "arma.op" estiver correta
                          }}}
```

```
                arma.comp <- sample(c("pedra","papel",
"tesoura","spock","lagarto"), 1)# Sorteia aleatóriamente por 1 vez e salva
no objeto "arma.comp"
                partidas[(n.partidas+1),2] <- arma.comp              #
Transfere "arma.comp" para a 2ª coluna, linha "i" de "partidas"
                         resultado <- tab.resul[arma.op,arma.comp]
# Procura os dois objetos na matriz "tab.resul"
                      if(resultado != "empate") break()
# Se "resultado" é diferente de "empate", quebra o ciclo de repetição
                }}}}

# Se resultado for ganha:
if(resultado == "ganha")
    {
    partidas[(n.partidas+1),3] <- resultado           # Acrescenta a
palavra à 3ª coluna, última linha da tabela "partidas"
    cat(cyan$bold("PARABÉNS! VOCÊ É O VENCEDOR!\n "))# Mostra a frase em cor
"cyan" e negrito
    tab.pontos[1,1] <- (tab.pontos[1,1]+1)            # Acrescenta 1 ponto
na 1ª coluna de "tab.pontos"
    print(sample(c(vitoria, vitoria.2),1))            # Reproduz
aleatoriamente uma das imagens salvas em "vitoria" e "vitoria.2"
    }



# Se o resultado for perde:
if(resultado == "perde")
    {
    partidas[(n.partidas+1),3] <- resultado                  # Acrescenta a
palavra à 3ª coluna, última linha da tabela "partidas"
    cat(magenta$bold("ESSA FOI POR POUCO... MAIS SORTE DA PRÓXIMA VEZ\n")) #
Mostra a frase em cor "magenta" e negrito
    tab.pontos[1,2] <- (tab.pontos[1,2]+1)                   # Acrescenta 1
ponto na 2ª coluna de "tab.pontos"
    print(sample(c(derrota, derrota.2),1))                   # Reproduz
aleatoriamente uma das imagens salvas em "derrota" e "derrota.2"
    }
}

cat(red("\n---- Tabela de pontos ----\n"))      # Reproduz a mensagem em
vermelho
print (tab.pontos)                              # Mostra a tabela de pontos
"tab.pontos"
cat(red("\n---- Tabela de jogadas ----\n"))     # Reproduz a mensagem em
vermelho
print(partidas)                                 # Mostra a tabela de jogadas
de "partidas"
}}}
```

From:
http://ecor.ib.usp.br/ - **ecoR**

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2019:alunos:trabalho_final:lidiayano:jokenpo

Last update: **2020/08/12 06:04**