

```

initialize global [I_AM_SERVER] to [false]

initialize global [I_HAVE_ANSWERED] to [false]

initialize global [DEF_STOP_ANSWER_WITH_REMOTE_ERROR] to ["STOP_ANSWER_WITH_REMOTE_ERROR"]

initialize global [DEF_STOP_ANSWER_WITH_REMOTE_OK] to ["STOP_ANSWER_WITH_REMOTE_OK"]

initialize global [DEF_START_PLAY] to ["START_PLAY"]

initialize global [MAX_TIME] to [10]

initialize global [GOOD_ANSWER] to [“”]

? initialize global [I_AM_HOST] to [true]

initialize global [IS_SINGLE_PLAYER] to [true]

initialize global [ALL_QUESTIONS] to [create empty list]

initialize global [LIST_IND_ANSWERED] to [create empty list]

initialize global [LIST_POSSIBLE_ANSWERS] to [create empty list]

initialize global [ITEM_CHOSEN] to [create empty dictionary]

initialize global [DEF_SERVER_CHOSEN] to [“SERVER_CHOSEN”]

initialize global [DEF_CMD_ANSWER] to [“AI”]

initialize global [TIME_INTERVAL] to [1000]

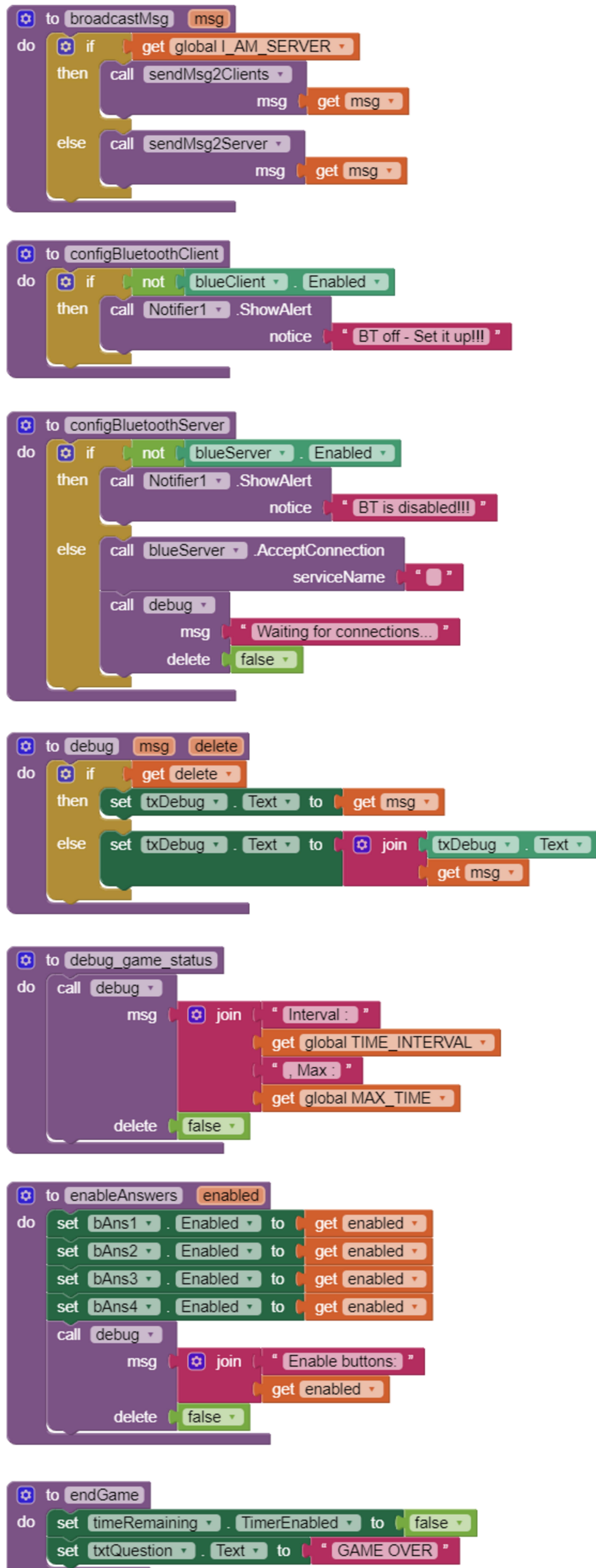
[? to actionStartPlaying]
do
  set [lyPlay v.].Visible to [true]
  set [lyWaiting v.].Visible to [false]
  set [lyConfig v.].Visible to [false]
  timeRemaining v. TimerEnabled v. true
  set [timeRemaining v.].TimerInterval to [get global TIME_INTERVAL]
  set [txTime v.].Text to [get global MAX_TIME]
  if [get global I_AM_HOST]
    then
      call [webJSON v.].ClearCookies
      call [webJSON v.].Get
      call [debug_game_status v.]

```

```

[? to actionWaiting]
do
  set [lyWaiting v.].Visible to [true]
  set [lyPlay v.].Visible to [false]
  set [lyConfig v.].Visible to [false]

```



```

when green flag clicked
  to getColorAnswer [answer]
    result initialize local [color] to [red]
    in do if [compare texts v [get answer] ≠ [get global GOOD_ANSWER]] then [set color v to [red]]
      result [get color]

```

```

when green flag clicked
  to getIndQuestion2Answer
    result initialize local [ind] to [0]
    in do if [length of list [list] < [get global LIST_IND_ANSWERED]] < [length of list [list] < [get global ALL_QUESTIONS]]
      then while test [true]
        do [set ind v to [random integer from [1] to [length of list [list]] < [get global ALL_QUESTIONS]]]
          if [not [is in list? [thing] [list]]] then [get ind]
            list [get global LIST_IND_ANSWERED]
          then [break]
        add items to list [list] [get global LIST_IND_ANSWERED]
        item [get ind]
        set [txIndChosen] . [Text] to [get ind]
      else call [endGame]
    result [get ind]

```

```

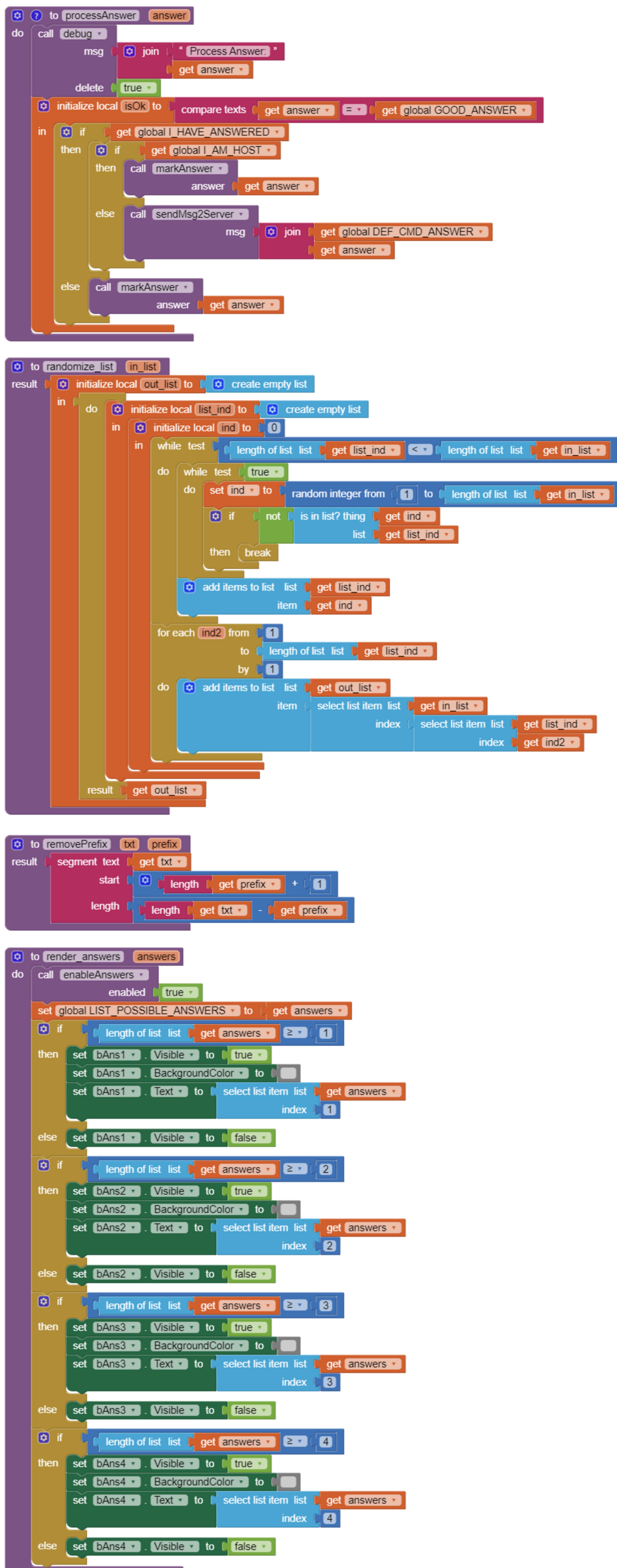
when green flag clicked
  to getMyAnswer [ind]
    result select list item [list] [get global LIST_POSSIBLE_ANSWERS]
    index [get ind]

```

```

when green flag clicked
  to markAnswer [answer]
    do call [enableAnswers]
      enabled [false]
      initialize local [isOk] to [if [compare texts v [get answer] = [get global GOOD_ANSWER]] then [get global I_HAVE_ANSWERED] else [not [get global I_HAVE_ANSWERED]]]
      in call [debug]
        msg [join "isOk? " [get isOk]]
        delete [false]
      initialize local [incPoints] to [1]
      in if [not [get isOk]] then [set incPoints v to [-1]]
      set [txPoints] . [Text] to [txPoints] . [Text] + [get incPoints]
      initialize local [incTimeInerval] to [0]
      in if [not [get isOk]] then [set incTimeInerval v to -100]
      set global TIME_INTERVAL to [global TIME_INTERVAL] + [get incTimeInerval]
      initialize local [incMaxTime] to [0]
      in if [not [get isOk]] then [set incMaxTime v to -1]
      set global MAX_TIME to [global MAX_TIME] + [get incMaxTime]
      set [txTime] . [Text] to [get global MAX_TIME]
      set (timeRemaining . TimerInterval) to [get global TIME_INTERVAL]
      call [debug_game_status]
      if [get global I_HAVE_ANSWERED] and [not [get global IS_SINGLE_PLAYER]] then call [broadcastMsg]
        msg [join [get global DEF_CMD_ANSWER] [get answer]]
      if [get global I_AM_HOST] then set [timerShowQuestion] . [TimerEnabled] to [true]

```



```

when green flag clicked
  do
    if blueClient . IsConnected
      then call blueClient . SendText
        text get msg

```

```

when green flag clicked
  do
    if blueServer . IsConnected
      then call blueServer . SendText
        text get msg

```

```

when green flag clicked
  do
    set global I_HAVE_ANSWERED to false
    if is empty get item
      then initialize local ind to call getIndQuestion2Answer
        in if get ind > 0
          then set item to select list item list get global ALL_QUESTIONS
            index get ind
            call debug
              msg join " Chosen question # "
                get ind
            delete false
            if get global I_AM_SERVER
              then call sendMsg2Clients
                msg get item
        end
    end
    if not is empty get item
      then call debug
        msg get item
        delete true
        set txtQuestion . Text to get value for key " q "
          in dictionary get item
          or if not found " ERROR QUESTION "
        if is key in dictionary? key " a "
          dictionary get item
          then set item to make a dictionary key " t " value
            if compare texts " T " = get value for key " a "
              in dictionary get item
              or if not found " not found "
              then " Verdadero "
              else " Falso "
            end
            key " f " value make a list if compare texts " T " =
              get value for key " a "
              in dictionary get item
              or if not found " not found "
              then " Falso "
              else " Verdadero "
            end
        end
        set global GOOD_ANSWER to get value for key " t "
          in dictionary get item
          or if not found " ERROR ANSWER "
        initialize local answers to copy list list get value for key " f "
          in dictionary get item
          or if not found " not found "
        in add items to list list get answers
          item get global GOOD_ANSWER
        set answers to call randomize_list in list get answers
        call render_answers
          answers get answers
        debug
          # answers :
            answers
            false
        debug
          Show question
            false

```



when blueServer .ConnectionAccepted

do

- set [timerReceiver v].TimerEnabled to true
- call [debug v]
 - msg "I am the server"
 - delete false
- set [global I_AM_SERVER v] to true
- set [global IS_SINGLE_PLAYER v] to false
- call [sendMsg2Clients v]
 - msg get [global DEF_SERVER_CHOSEN v]

when bAns1 .Click

do

- set [global I_HAVE_ANSWERED v] to true
- initialize local [answer v] to call [getMyAnswer v]
 - ind 1
- in set [bAns1 v].BackgroundColor to call [getColorAnswer v]
 - answer get [answer v]
- call [processAnswer v]
 - answer get [answer v]

when bAns2 .Click

do

- set [global I_HAVE_ANSWERED v] to true
- initialize local [answer v] to call [getMyAnswer v]
 - ind 2
- in set [bAns2 v].BackgroundColor to call [getColorAnswer v]
 - answer get [answer v]
- call [processAnswer v]
 - answer get [answer v]

when bAns3 .Click

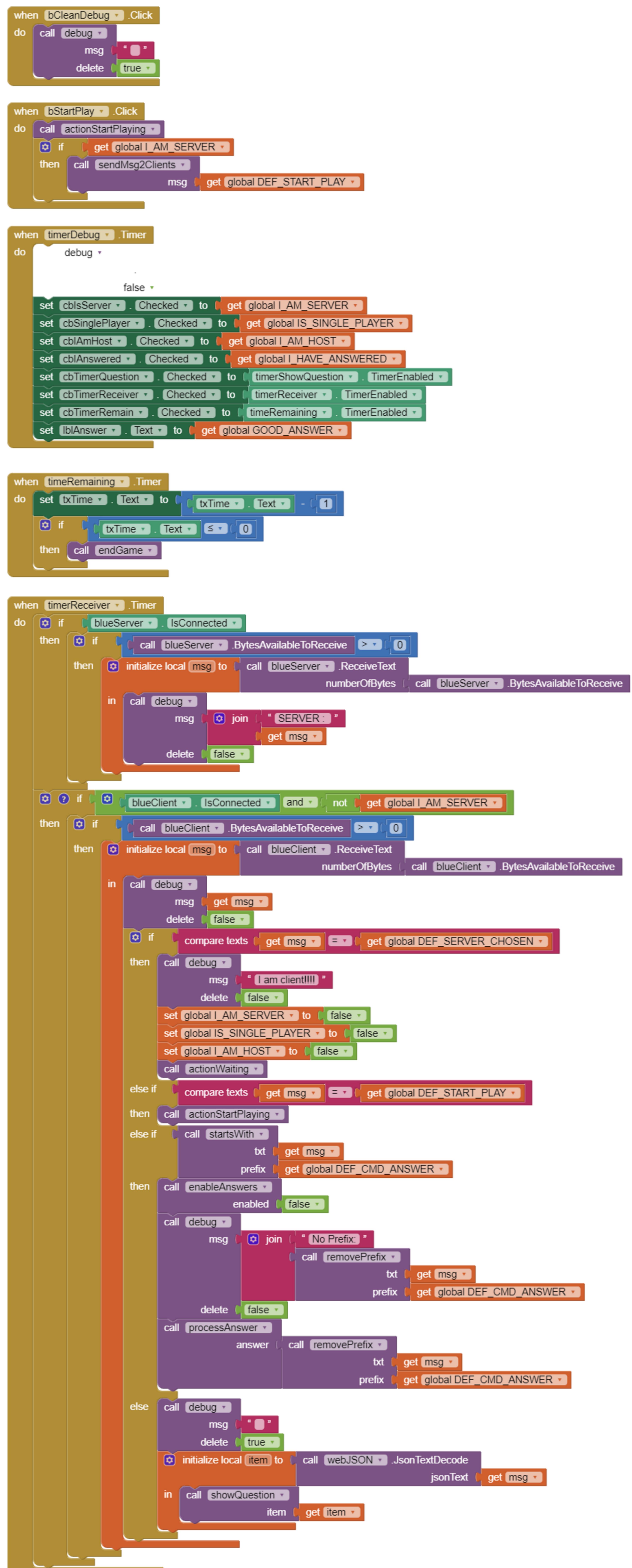
do

- set [global I_HAVE_ANSWERED v] to true
- initialize local [answer v] to call [getMyAnswer v]
 - ind 3
- in set [bAns3 v].BackgroundColor to call [getColorAnswer v]
 - answer get [answer v]
- call [processAnswer v]
 - answer get [answer v]

when bAns4 .Click

do

- set [global I_HAVE_ANSWERED v] to true
- initialize local [answer v] to call [getMyAnswer v]
 - ind 4
- in set [bAns4 v].BackgroundColor to call [getColorAnswer v]
 - answer get [answer v]
- call [processAnswer v]
 - answer get [answer v]



```

when timerShowQuestion .Timer
do set timerShowQuestion .TimerEnabled to false
call showQuestion
item " "

```

```

when scrPlay .Initialize
do set lyPlay .Visible to false
set lyConfig .Visible to true
call configBluetoothServer

```

```

when scrPlay .ErrorOccurred
component functionName errorNumber message
do call debug
msg join " Server Error: "
get component
" "
get functionName
" "
get errorNumber
" "
get message
delete false

```

```

when IpConnect .BeforePicking
do set IpConnect .Elements to blueClient . AddressesAndNames
call debug
msg blueClient . AddressesAndNames
delete false

```

```

when IpConnect .AfterPicking
do if call blueClient .Connect
address IpConnect . Selection
then set timerReceiver . TimerEnabled to true
call blueClient .SendText
text " Client connected!!! "
call debug
msg " Client connected!!! "
delete false

```

```

when webJSON .GotText
url responseCode responseType responseContent
do set global ALL_QUESTIONS to call webJSON .JsonTextDecodeWithDictionaries
jsonText get responseContent
set txNumberQ . Text to length of list list get global ALL_QUESTIONS
set timerShowQuestion . TimerEnabled to true

```