

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 "A|"

initialize global TIME_INTERVAL to 1000

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

```
to actionWaiting
do
  set lyWaiting . Visible to true
  set lyPlay . Visible to false
  set lyConfig . Visible to false
```

```
to broadcastMsg msg
do
  if get global I_AM_SERVER
  then
    call sendMsg2Clients
      msg get msg
  else
    call sendMsg2Server
      msg get msg
```

```
to configBluetoothClient
do
  if not blueClient.Enabled
  then
    call Notifier1.ShowAlert
      notice "BT off - Set it up!!!"
```

```
to configBluetoothServer
do
  if not blueServer.Enabled
  then
    call Notifier1.ShowAlert
      notice "BT is disabled!!!"
  else
    call blueServer.AcceptConnection
      serviceName ""
    call debug
      msg "Waiting for connections..."
    delete false
```

```
to debug msg delete
do
  if get delete
  then
    set txDebug.Text to get msg
  else
    set txDebug.Text to join txDebug.Text
      get msg
```

```
to debug_game_status
do
  call debug
    msg join "Interval : "
      get global TIME_INTERVAL
      ", Max : "
      get global MAX_TIME
  delete false
```

```
to enableAnswers enabled
do
  set bAns1.Enabled to get enabled
  set bAns2.Enabled to get enabled
  set bAns3.Enabled to get enabled
  set bAns4.Enabled to get enabled
  call debug
    msg join "Enable buttons: "
      get enabled
  delete false
```

```
to endGame
do
  set timeRemaining.TimerEnabled to false
  set txtQuestion.Text to "GAME OVER"
```

```

to getColorAnswer answer
result
  initialize local color to green
  in
    do
      if compare texts get answer ≠ get global GOOD_ANSWER
      then set color to red
      result get color

```

```

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 to random integer from 1 to length of list list get global ALL_QUESTIONS
          if not is in list? thing 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

```

```

to getMyAnswer ind
result
  select list item list get global LIST_POSSIBLE_ANSWERS
  index get ind

```

```

to markAnswer answer
do
  call enableAnswers
  enabled false
  initialize local isOk to
    if compare texts 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? "
    delete false
    get isOk
    initialize local incPoints to 1
    in
      if not get isOk
      then set incPoints to -1
    set txPoints . Text to txPoints . Text + get incPoints
    initialize local incTimeInterval to 0
    in
      if not get isOk
      then set incTimeInterval to -100
    set global TIME_INTERVAL to get global TIME_INTERVAL + get incTimeInterval
    initialize local incMaxTime to 0
    in
      if not get isOk
      then set incMaxTime to -1
    set global MAX_TIME to get global MAX_TIME + get incMaxTime
    set txTime . Text to get global MAX_TIME
  set timeRemaining . TimeInterval 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

```

```

to processAnswer answer
do
  call debug
  msg join "Process Answer:"
  delete true
  initialize local isOk to compare texts get answer = get global GOOD_ANSWER
  in if get global I_HAVE_ANSWERED
  then if get global I_AM_HOST
  then call markAnswer
  answer get answer
  else call sendMsg2Server
  msg join get global DEF_CMD_ANSWER
  get answer
  else call markAnswer
  answer get answer

```

```

to randomize_list in_list
result
  initialize local out_list to create empty list
  in do initialize local list_ind to create empty list
  in initialize local ind to 0
  in while test length of list list get list_ind < length of list list get in_list
  do while test true
  do set ind to random integer from 1 to length of list list get in_list
  if not is in list? thing get ind
  list get list_ind
  then break
  add items to list list get list_ind
  item get ind
  for each ind2 from 1
  to length of list list get list_ind
  by 1
  do add items to list list get out_list
  item select list item list get in_list
  index select list item list get list_ind
  index get ind2
  result get out_list

```

```

to removePrefix txt prefix
result
  segment text get txt
  start length get prefix + 1
  length length get txt - get prefix

```

```

to render_answers answers
do
  call enableAnswers
  enabled true
  set global LIST_POSSIBLE_ANSWERS to get answers
  if length of list list get answers ≥ 1
  then set bAns1 . Visible to true
  set bAns1 . BackgroundColor to
  set bAns1 . Text to select list item list get answers
  index 1
  else set bAns1 . Visible to false
  if length of list list get answers ≥ 2
  then set bAns2 . Visible to true
  set bAns2 . BackgroundColor to
  set bAns2 . Text to select list item list get answers
  index 2
  else set bAns2 . Visible to false
  if length of list list get answers ≥ 3
  then set bAns3 . Visible to true
  set bAns3 . BackgroundColor to
  set bAns3 . Text to select list item list get answers
  index 3
  else set bAns3 . Visible to false
  if length of list list get answers ≥ 4
  then set bAns4 . Visible to true
  set bAns4 . BackgroundColor to
  set bAns4 . Text to select list item list get answers
  index 4
  else set bAns4 . Visible to false

```

```

to sendMsg2Server msg
do
  if blueClient . isConnected
  then call blueClient . SendText
        text get msg

```

```

to sendMsg2Clients msg
do
  if blueServer . isConnected
  then call blueServer . SendText
        text get msg

```

```

to showQuestion item
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
          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
      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 "
            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 "
        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
          list get value for key " f "
          in dictionary get item
          or if not found " not found "
        in
          add items to list 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

```

```
to startsWith txt prefix
result
  compare texts get prefix = segment text get txt
  start 1
  length length get txt
```

```
when blueServer .ConnectionAccepted
do
  set timerReceiver .TimerEnabled to true
  call debug
  msg " I am the server "
  delete false
  set global I_AM_SERVER to true
  set global IS_SINGLE_PLAYER to false
  call sendMsg2Clients
  msg get global DEF_SERVER_CHOSEN
```

```
when bAns1 .Click
do
  set global I_HAVE_ANSWERED to true
  initialize local answer to call getMyAnswer ind 1
  in
    set bAns1 .BackgroundColor to call getColorAnswer answer get answer
    call processAnswer answer get answer
```

```
when bAns2 .Click
do
  set global I_HAVE_ANSWERED to true
  initialize local answer to call getMyAnswer ind 2
  in
    set bAns2 .BackgroundColor to call getColorAnswer answer get answer
    call processAnswer answer get answer
```

```
when bAns3 .Click
do
  set global I_HAVE_ANSWERED to true
  initialize local answer to call getMyAnswer ind 3
  in
    set bAns3 .BackgroundColor to call getColorAnswer answer get answer
    call processAnswer answer get answer
```

```
when bAns4 .Click
do
  set global I_HAVE_ANSWERED to true
  initialize local answer to call getMyAnswer ind 4
  in
    set bAns4 .BackgroundColor to call getColorAnswer answer get answer
    call processAnswer answer get answer
```

```
when bCleanDebug .Click
do
  call debug
  msg " "
  delete true
```

```
when bStartPlay .Click
do
  call actionStartPlaying
  if get global I_AM_SERVER
  then call sendMsg2Clients
  msg get global DEF_START_PLAY
```

```
when timerDebug .Timer
do
  debug
  false
  set cbIsServer .Checked to get global I_AM_SERVER
  set cbSinglePlayer .Checked to get global IS_SINGLE_PLAYER
  set cbAmHost .Checked to get global I_AM_HOST
  set cbAnswered .Checked to get global I_HAVE_ANSWERED
  set cbTimerQuestion .Checked to timerShowQuestion .TimerEnabled
  set cbTimerReceiver .Checked to timerReceiver .TimerEnabled
  set cbTimerRemain .Checked to timeRemaining .TimerEnabled
  set lblAnswer .Text to get global GOOD_ANSWER
```

```
when timeRemaining .Timer
do
  set txtTime .Text to txtTime .Text - 1
  if txtTime .Text ≤ 0
  then call endGame
```

```
when timerReceiver .Timer
do
  if blueServer .IsConnected
  then
    if call blueServer .BytesAvailableToReceive > 0
    then
      initialize local msg to call blueServer .ReceiveText
      numberOfBytes call blueServer .BytesAvailableToReceive
      in call debug
      msg join " SERVER: "
      get msg
      delete false
    if blueClient .IsConnected and not get global I_AM_SERVER
    then
      if call blueClient .BytesAvailableToReceive > 0
      then
        initialize local msg to call blueClient .ReceiveText
        numberOfBytes call blueClient .BytesAvailableToReceive
        in call debug
        msg get msg
        delete false
        if compare texts get msg = get global DEF_SERVER_CHOSEN
        then
          call debug
          msg " I am client!!!! "
          delete false
          set global I_AM_SERVER to false
          set global IS_SINGLE_PLAYER to false
          set global I_AM_HOST to false
          call actionWaiting
        else if compare texts get msg = get global DEF_START_PLAY
        then
          call actionStartPlaying
        else if call startsWith
        txt get msg
        prefix get global DEF_CMD_ANSWER
        then
          call enableAnswers
          enabled false
          call debug
          msg join " No Prefix: "
          call removePrefix
          txt get msg
          prefix get global DEF_CMD_ANSWER
          delete false
          call processAnswer
          answer call removePrefix
          txt get msg
          prefix get global DEF_CMD_ANSWER
        else
          call debug
          msg " "
          delete true
          initialize local item to call webJSON .JsonTextDecode
          jsonText get msg
          in call showQuestion
          item get item
```

```
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

```