```
#!/usr/bin/tclsh

if {![info exists env(SPACKLE_HOME)]} {
  set spackleHome ./
} else {
  set spackleHome $env(SPACKLE_HOME)
}

lappend auto_path $spackleHome


package require Spackle

proc createObject {oname} {
  namespace eval $oname {
    variable thisComputer ""
    variable logfile [pwd]/taggame.log

    proc wakeup {} {
      after idle [this]::playGame
    }

    proc playGame {} {
      set others [::comm::comm send $::Spackle::AgentSrvr::remoteInterp \
          ::Spackle::Portal::who]
      if {[llength $others] > 0} {
        # pick one at random
        array set aothers $others
        set names [array names aothers]
        set size [llength $names]
        set which [expr {int (rand()*$size)}]
        set name [lindex $names $which]
        set otherinterp $aothers($name)
        # call tag
        if {[catch {::comm::comm send $otherinterp ${name}::tag} results]} {
          logit "[this]: I missed $name!"
        } else {
          logit "[this]: Your out $name!"
        }
      } else {
        logit "[this]: No one to tag!"
      }
      # add self to registry
      set j [::comm::comm send $::Spackle::AgentSrvr::remoteInterp \
        ::Spackle::Portal::registerMe [this] [::comm::comm self]]
      # Let others have a chance
      set waitTime [expr {int (rand()*3000) + 2000}]
      #set waitTime 5000
      after $waitTime [this]::moveOn
    }


    proc moveOn {} {
      variable computers
      variable thisComputer
      # remove self from registry
      ::comm::comm send $::Spackle::AgentSrvr::remoteInterp \
        ::Spackle::Portal::unregisterMe [this]
      # pick random machine
      set size [llength $computers]
      while {1} {
        flush stdout
        set which [expr {int (rand()*$size)}]
        set machine [lindex $computers $which]
        if {$thisComputer != $machine} {
          break;
        }
        #If we get here, then we chose ourself, try again
      }
      set thisComputer $machine
      # move
      # We can do this becuase it only set a flag, and the server kills us at
      #   its leasure
      ::Spackle::AgentSrvr::die
      ::Spackle::Portal::phase [this] [this] $machine
    }

    proc setMachines {mList} {
      variable computers
      set computers $mList
    }
    proc this {} {
      return [namespace current]
    }
    proc logit {msg} {
      variable thisComputer
      variable logfile
      catch {
        set fd [open $logfile "a"]
        puts $fd "[format %-30s *$thisComputer*] $msg"
        close $fd
      }
    }
    proc tag {} {
      ::comm::comm send $::Spackle::AgentSrvr::remoteInterp \
        ::Spackle::Portal::unregisterMe [this]
      logit "[this]: tagged, I'm out"
      after 1000 {::Spackle::AgentSrvr::die}
    }
  }
}

proc main {} {
  # argv 1 is number of tag objects, rest if list of machines
  set computers [lrange $::argv 1 end]
  set count [lindex $::argv 0]
  set fd [open [pwd]/taggame.log "w"]
  close $fd
  for {set k 0} {$k < $count} {incr k} {

    createObject ::tagobject$k

    ::tagobject${k}::setMachines $computers
    set size [llength $computers]
    set which [expr {int (rand()*$size)}]
    set machine [lindex $computers $which]
    set ::tagobject${k}::thisComputer $machine

    puts "phasing to $machine..."

    if {[catch {set interp [::Spackle::Portal::phase  ::tagobject$k \
        ::tagobject$k $machine]} result]} {
      puts $result
      puts $errorInfo
      puts "Done."
      exit 1
    }

  }

  puts "Done."
}

main
```