movie script (Internal)

global gPurchaseList
--This list will keep track of the prices of all merchandise added to the "shopping cart."

on prepareMovie
  installMenu "MENU"
  -- The cast member "MENU" is installed, allowing users to quit or start over.
end

on resetList
  set gPurchaseList = []
  put gPurchaseList
  -- This handler, called in frame 1, empties the list and relays this fact in the message window.
end

on sum whateverList
  -- Since "whateverList" acts as a placeholder for a list sent by another script using the command
  -- "sum" (in this case, the "find total" script) this handler can be used with any linear list.
  repeat with x in whateverList
    -- This repeat loop puts each item in the place of 'x' until all list items have been added up.
    set purchaseTotal = purchaseTotal + x
    put "add next price -- " && x && " -- and the total is now up to " && purchaseTotal
    -- The message window shows what price is currently being added and provides a running total.
  end repeat
  put "the final total for your purchases is " & purchaseTotal
  -- The message window reveals the final total once the repeat loop is finished.
  return purchaseTotal
  -- The sum of all list items is returned to the "find total" script from the handler was called.
end sum

Send Price (Internal)

on mouseUp
  global gPurchaseList
  -- declare your list just like you would any other global variable
  set the visible of sprite (the currentSpriteNum) = 0
  -- this item disappears from the screen when you put it in your shopping cart
  set newPrice = value(the name of the member of sprite (the currentSpriteNum))
  -- convert the name of this cast member from a string into a value
  put "the price of this item is " & newPrice
  -- watch in the message window to see what the price is
  add gPurchaseList, newPrice
  -- add the price of the new item to your list of prices
  put gPurchaseList
  -- keep track of your list in the message window
end
on mouseUp
  global gPurchaseList
  -- declare your list just like you would any other global variable
  go marker(1)
  -- go to the next marker, where the user will be told whether she wins or loses
  set randomMoneyNumber = random(3) + 9
  -- Director randomly picks a number 10 and 12.
  set moneyMemberName = the name of member randomMoneyNumber
  -- This randomly chosen number is used to refer to one of the three cast members
  -- that indicates on screen how much cash the user has available.
  set the visible of sprite randomMoneyNumber = 1
  -- You will notice that cast member 10 is placed in sprite channel 10, cast member
  -- 11 is placed in sprite channel 11, etc. This makes it easy to use the same number
  -- to refer to a cast member and to make the chosen dollar value visible on stage.
  set randomTotal = value(moneyMemberName)
  -- Convert the name of the cast member into a value that is compared to the
  -- user's purchase total. This works only because the cast members are named
  -- using the dollar amount to which they refer. Make sure no cast member
  -- names are repeated (e.g. no two items can cost the same or represent the
  -- same dollar amount in the user's wallet).
  put "The amount of money randomly assigned to you is" && randomTotal
  -- The results of this randomization process is displayed in the message window.
  if sum (gPurchaseList) > randomTotal then
    set the visible of sprite 13 = 1
    put "the sum of your purchases is larger than the amount you have in your wallet"
    -- If the user spends more than the amount of cash she has available, then the
    -- "you lose" message comes up, and a similar message is displayed in the message
    -- window.
  else
    if sum (gPurchaseList) < (randomTotal/2) then
      set the visible of sprite 14 = 1
      put "the sum of your purchases is less than half of the amount you have in your wallet"
      -- If the user spends less than half of the available funds, then a "frugal shopper"
      -- message comes up, and a similar message is displayed in the message window.
    else
      set the visible of sprite 15 = 1
      put "you've stayed within your budget, but spent more than half your assets."
      -- If the user spends more than half her budget but does not exceed her budget,
      -- then a "you win" message comes up, and a similar message is displayed in
      -- the message window.
    end if
  end if
updateStage
  -- The screen is refreshed to show the win/lose message.
end