SAMPLE SOURCE CODE:

```
import viz
import vizact
import vizshape
import viztask
import sys
import vizinput
import vizinfo
import vizmat
import projector

viz.go()

#Add ARToolkit extension
ar = viz.add('artoolkit.dle')

#Importing Web Camera
camera = ar.addWebCamera()

#Setting Board as Reference Marker
board = camera.addMatrixMarker(6,width=1000)
camera.setGlobalMarker(board)

####  GUI'S

#Create GUI for the 'start' button.
startButton = viz.addButtonLabel('START')
startButton.setPosition(.94,.04)
startButton.enable

###  MESSAGES

###Message to inform user to click start to begin.
text_2M = viz.addText('Click START to begin',viz.SCREEN )
text_2M.fontSize(30)
text_2M.setPosition(.35, .97)
text_2M.color(viz.GREEN)

#Message to inform user to position the 1st Guide Wall.
text_2M1 = viz.addText('Use image to the place the first "Guide Wall" ; Mount with screws', viz.SCREEN )
text_2M1.fontSize(30)
text_2M1.setPosition(.15, .97)
text_2M1.color(viz.GREEN)

#Message to inform user to position the 2nd Guide Wall.
text_2M2 = viz.addText('Use image to the place second "Guide Wall" ; Mount with screws', viz.SCREEN )
text_2M2.fontSize(30)
text_2M2.setPosition(.15, .97)
text_2M2.color(viz.GREEN)

#Message to inform user 1st Guide Wall is properly positioned.
text_2D1 = viz.addText('First Guide Wall is properly placed, mount with screws and left click', viz.SCREEN )
text_2D1.fontSize(38)
text_2D1.setPosition(.08,.57)
text_2D1.color(viz.RED)
```

```
#Message to inform user 2nd Guide Wall is properly positioned.
text_2D2 = viz.addText('Second Guide Wall is properly placed, mount with screws and left click', viz.SCREEN )
text_2D2.fontSize(38)
text_2D2.setPosition(.06,.57)
text_2D2.color(viz.RED)

###  GLOBAL VARIABLES & Matrix Markers

#Importing virtual 3D model images.
guideWall1 = viz.add('morton_guidewall.wrl')
guideWall2 = viz.add('morton_guidewall.wrl')

#Declaring Matrix Markers.
barA = camera.addMatrixMarker(0,width=25)
barB = camera.addMatrixMarker(1,width=25)
### Functions for Bars

# 1st Guide Wall Function: position, orientation, transparency, and size declared.
def guideWall_A():
  a = guideWall1
  a.setPosition(-7.1,-.5,.7)
  a.setEuler(92,0,0)
  a.alpha( .8)
  a.setScale(.7,.6,.6)
  barA_pos = barA.getPosition(viz.ABS_GLOBAL)
  if round(barA_pos[0],2) > 1.14  and round(barA_pos[0],2) < 1.10 and round(barA_pos[1],2) > 22.50 and round(barA_pos[1],2)
< 22.43 and round(barA_pos[2],2) > -24.00 and round(barA_pos[2],2) < -22.06:
    text_2M1.visible(viz.OFF)
    a.remove()
    text_2D1.visible(viz.ON)
  else:
    print round(barA_pos[0],2),round(barA_pos[1],2),round(barA_pos[2],2)
#Timer for 1st Guide Wall function; disabled until called.
gwA_timer = vizact.ontimer2(1,viz.FOREVER,guideWall_A)
gwA_timer.setEnabled(viz.OFF)


#2nd Guide Wall Function: position, orientation, transparency, and size declared.
def guideWall_B():
 b = guideWall2
 b.setPosition(-10.8,-.5,.7)
 b.setEuler(92,0,0)
 b.alpha(.8)
 b.setScale(.7,.6,.6)
 barB_pos = barB.getPosition(viz.ABS_GLOBAL)
 if round(barB_pos[0],2) > -0.36 and round(barB_pos[0],2) < -0.20 and round(barB_pos[1],2) > -0.16 and round(barB_pos[1],2) <
-0.10 and round(barB_pos[2],2) > -0.29 and round(barB_pos[2],2) < -0.23:
  text_2M2.visible(viz.OFF)
  b.remove()
  text_2D2.visible(viz.ON)
 else:
  print round(barB_pos[0],2),round(barB_pos[1],2),round(barB_pos[2],2)

#Timer for 2nd Guide Wall function; disabled until called.
gwB_timer = vizact.ontimer2(1,viz.FOREVER,guideWall_B)
gwB_timer.setEnabled(viz.OFF)
```

```
### Functions for Timers

#1st Guide Wall timer is enabled; remaining timers are disabled.
def guideWall_A_timer():
 gwA_timer.setEnabled(viz.ON)
 gwB_timer.setEnabled(viz.OFF)

#2nd Guide Wall timer is enabled; remaining timers are disabled.
def guideWall_B_timer():
 gwA_timer.setEnabled(viz.OFF)
 gwB_timer.setEnabled(viz.ON)

####  Execute Program

def Execute():
 while True:
  #Only 'START' message visible.
  text_2M.visible(viz.ON)
  text_2M1.visible(viz.OFF)
  text_2M2.visible(viz.OFF)

  text_2D1.visible(viz.OFF)
  text_2D2.visible(viz.OFF)

  #All images invisible until the program starts.
  guideWall1.visible(viz.OFF)
  guideWall2.visible(viz.OFF)

  #Yields sequence until 'START' button is clicked.
  yield viztask.waitButtonDown(startButton)

  #Running guideWall_A
  startButton.remove()                    #Removes START button.
  yield viztask.waitMouseUp(viz.MOUSEBUTTON_LEFT)  #Yields sequence until mouse is click.
  text_2M.visible(viz.OFF)                #Removes START message.
  text_2M1.visible(viz.ON)                #Turn on message for 1st GuideWall.
  guideWall1.visible(viz.ON)              #1st GuideWall image visible.
  yield guideWall_A_timer()               #Activate timer for 1st Guidewall.
  yield guideWall_A()                     #Runs 1st GuideWall function.
  yield viztask.waitMouseUp(viz.MOUSEBUTTON_LEFT)  #Mouse click terminates function.

  #Running guideWall_B
  text_2D1.visible(viz.OFF)               #1st guideWall message off.
  text_2M2.visible(viz.ON)                #2nd guideWall message on.
  guideWall2.visible(viz.ON)              #2nd guideWall image visible.
  yield guideWall_B_timer()               #Activate timer for 2nd guidewall
  yield guideWall_B()                     #Runs 2nd guideWall function.
  yield viztask.waitMouseUp(viz.MOUSEBUTTON_LEFT)  #Mouse click terminates function
  text_2D2.visible(viz.OFF)               #2nd guideWall message off.

  #Alerts the student assembly is complete
  yield vizinput.message('Congratulations You Have Successfully Connected A "6 Bar Quick Return,"!!')
  yield vizinput.message('Click OK!!')
  #Exit program
  yield viz.quit()
viztask.schedule( Execute() )
```