

```

# File generated automatically by WorldViz graphical setup tool 3.1
# Setup option Input    --> Manufacturer: Desktop    Type: Mouse as right hand
# Setup option Tracker  --> Manufacturer: Ascension   Type: MotionStar Head+RHand
# Setup option Display  --> Manufacturer: TriVisio HMD Type: TriVisio 40dFOV Stereo
# Setup option Avatar   --> Manufacturer: Generic     Type: No Avatar

# This file must be loaded by the correct version of viztracker.py included with Vizard
import viz
viz.requireVersion("3.17.0000")

# Source code that defines the composite. This is user editable but changes will be lost if
# this file is regenerated
import vizact
#import vizuniverse as VU
import vizuniverse_drivernodefix as VU
import __main__
import hand

PORT_5DT_USB = 1 #Connect to first USB device
glove = viz.addSensor('5dt.dls')
data = glove.getData() #Get glove data

PORT_5DT_USB = 0 #Connect to first USB device
glove1 = viz.add('5dt.dls')
data1 = glove1.getData() #Get glove data

class VUJoystickButtons(VU.VUJoystickStandard):
    """This generic class is a wrapper around a standard vizjoy joystick, and other types
    of input devices can also implement a similar interface"""
    wii = viz.add('wiimote.dle')
    #Connect to first available wiimote
    wiimote = wii.addWiimote()

    def isButtonDown(self, num):
        """Return button information from the joystick, note that to be consistent we number
        from 0 although vizjoy starts from 1"""
        if num < 0:
            return 0
        elif num==0:
            if self.wiimote.isButtonDown(self.wii.BUTTON_B):
                return True
        elif num==1:
            if self.wiimote.isButtonDown(self.wii.BUTTON_MINUS):
                return True
        else:
            return self.joydev.isButtonDown(num+1)

    def getRawX(self):
        """Return the X value of the joystick"""
        if self.wiimote.isButtonDown(self.wii.BUTTON_RIGHT): #index starts at 1
            return 1
        elif self.wiimote.isButtonDown(self.wii.BUTTON_LEFT): #index starts at 1
            return -1
        else:
            return 0

    def getRawY(self):

```

```

        """Return the Y value of the joystick"""
        if self.wiimote.isButtonDown(self.wii.BUTTON_DOWN): #index starts at 1
            return 1
        elif self.wiimote.isButtonDown(self.wii.BUTTON_UP): #index starts at 1
            return -1
        else:
            return 0

# Create a custom composite that handles tracking, display, and input devices all together
def createCustomComposite(id=0):
    # ---- Trackers ----
    # Initialize an empty composite object to store all the trackers
    # The composite.storeTracker() method is used to combine the individual trackers for the
    # user's body within the composite

    ascension = viz.add('ascension.dle')
    sensors = ascension.addTrakStar()

    composite = VU.VUCompositeTrackers()
    composite.addDriverNode(viz.addGroup(pos=[0, 1, 0])) #origen offset
    composite.storeTracker (composite.HEAD, sensors[0])
    composite.storeTracker(composite.LHAND, sensors[2])
    composite.storeTracker(composite.RHAND, sensors[1])
    composite.setPosScale([1.0, 1.0, 1.0]) #movement scale

    #wandflyer = VU.VUTrackerWandFlyer(controller=VU.VUJoystickStandard(), tracker=sensors
    [1], speed=.02, xscale=1.0, yscale=1.0, zscale=1.0, oriscale=1.0, oriSteer=True, walkingMo
    de=True, keystrokes=['a','b','c','d','d','r'], buttonReset=None, buttonForward=None, butt
    onReverse=None, buttonFist=0, buttonUp=2, buttonDown=3, buttonPinch=4)
    wandflyer = VU.VUTrackerWandFlyer(controller=VUJoystickButtons(), tracker=sensors[1],
    speed=.02, xscale=1.0, yscale=1.0, zscale=1.0, oriscale=1.0, oriSteer=True, walkingMode=Tr
    ue, keystrokes=[None,'r',None,None,None,None], buttonReset=1, buttonForward=None, buttonR
    everse=None, buttonFist=None, buttonUp=None, buttonDown=None, buttonPinch=0)
    composite.addDriverNode(wandflyer)

    composite.getRawTracker(composite.HEAD).getLink().preEuler([0,0,0]) #yaw, pitch, roll
    composite.getRawTracker(composite.LHAND).getLink().preEuler([0,0,180]) #move forward .
    1 metre
    composite.getRawTracker(composite.RHAND).getLink().preEuler([0,0,180]) #move forward .
    1 metre

    # ---- Display ----
    viz.setOption ('viz.fullscreen', 1)
    #viz.setOption ('viz.display.width', 800) #rmc 23/10/14 for oculus
    #viz.setOption ('viz.display.height', 600) #rmc 23/10/14 for oculus

    #import nvis
    #nvis.HMD(34, 25) #horizontal stereo, 34 horizontal fov, 25 vertical fov
    #import trivisio
    #trivisio.HMD(44,35)

    #Oculus rift hmd
    import oculus
    oculus.HMD()

    # ---- Input ----
    #CYBERGLOVE
    #CyberGlove = viz.add('CyberGlove.dle')
    #lhand = CyberGlove.addCyberGlove()

```

```

#composite.createLeftHand (lhand,handSize=1.0)

#Specify USB port for 5DT Glove
#5DT Glove
PORT_5DT_USB = 1 #Connect to first USB device
glove = viz.addSensor('5dt.dls')
data = glove.getData() #Get glove data
composite.createLeftHand (glove,handSize=1.0)

PORT_5DT_USB = 0 #Connect to first USB device
glove1 = viz.add('5dt.dls')
data1 = glove1.getData() #Get glove data
composite.createRightHand (glove1,handSize=1.0)

#Other Hands
#composite.createRightHand(wandflyer.getHandSensor())

# ---- Avatar ----
composite.createAvatarNone()

# ---- Finalize Composite ----
composite.finishTrackers() # Build up internal links for all the tracking devices
composite.defineViewpoint() # Attach viewpoint to default location on the user

# ---- Extra Adjustments Editable By The User ----

# Note that the examples below will not work with LiveCharacters, only inputs with ind
ividual trackers
# composite.getRawTracker(composite.HEAD).getLink().postEuler([0,0,45]) # Apply 45
degree offset to the head tracker
# composite.getLeftHand().setScale([2,2,2]) # Apply sc
ale to the left hand representation
# composite.getLeftHand().alpha(0.0) # Make the
left hand invisible
# composite.getRawTracker(composite.LHAND).getLink().postTrans([0,-0.5,0]) # Lower th
e hand tracker 50 cm
# composite.setPosScale([2,1,2]) # Scale XZ
position of all trackers by a factor of 2x
# composite.setOriScale([0,0,0]) # Suppress
all orientation changes for all trackers

# The following examples are compatible with all kinds of composite objects, including
LiveCharacters
# composite.getMovableNode().setPosition([1.0,0.0,0.0]) # Move the
user and all trackers +1 unit along the X axis
# composite.defineViewpoint(offset=[0,0,0]) # Replace
above defineViewpoint() call, allows custom offset of viewpoint

# ---- Return Back Result ----
return composite

# If this script is run directly, then we should allow it to work for testing purposes (no
rmally viztracker.py loads this file in, and this file is not used standalone)
if __name__ == "__main__":
    print 'Manually overriding viztracker to test vizsetupcfg configuration'
    import viztracker
    viztracker.createCustomComposite = createCustomComposite
    viztracker.go()

```

```
viz.add('gallery.ive')  
PORT_5DT_USB = 1 #Connect to first USB device  
glove = viz.addSensor('5dt.dls')  
data = glove.getData() #Get glove data  
PORT_5DT_USB = 0 #Connect to first USB device  
glove1 = viz.addSensor('5dt.dls')  
data1 = glove1.getData() #Get glove data
```