```
from sys import path
path.append("/home/chfrag/src/cgal-python-0.9.1/cgal_package")

from visual import *
from CGAL import *

class mouseClick(object):
        """
        click = mouseClick(which_button=None)
        """
        def __init__(self, which_button=None):
                self.button = None
                self.pos = None
                self.pick = None
                self.get(which_button)

        def getEvent(self):
                if scene.mouse.events:
                        event = scene.mouse.getevent()
                        if event.press:
                                self.button = event.button
                                self.pos = event.pos
                                self.pick = event.pick

        def get(self, which_button):
                if which_button is not None:
                        while self.button <> which_button:
                                self.getEvent()
                else:
                        while self.button is None:
                                self.getEvent()

        def __str__(self):
                return "%s button pressed at position %s" % (self.button, self.pos)


class VPoint2d(object):
        """
        Holds the visual representation of a CGAL.Point_2() point ...
        """
        def __init__(self, point2d=None, canBeNone=False):
                self.vpoint2d = None
                self.point2d = point2d
                self.canBeNone = canBeNone

                self._color = None
                self._label = None

                if self.point2d is not None:
                        if self.label is not None:
                                self.repr = 'label'
                        else:
                                self.repr = 'sphere'
                else:
                        self.get()

        def get(self):
                # get a point using left or middle mouse buttons
                if self.canBeNone:
                        click = mouseClick()
                        if click.button == 'right':
                                # self.point2d remains None as set in __init__
                                return
                else:
                        click = mouseClick('left')

                # self.point2d gets a value
                self.point2d = Point_2(click.pos.x, click.pos.y)
                self.repr = 'sphere'
```



```
        def representation():
                doc = "Representation of VPoint2: visual.sphere or visual.label"

                def fget(self):
                        if self.vpoint2d.__class__ is sphere:
                                return 'sphere'
                        else:
                                return 'label'

                def fset(self, value):
                        if self.vpoint2d is not None:
                                self.vpoint2d.visible = False
                        vpos = (self.point2d.x(), self.point2d.y())
                        vcolor = self._color or color.white
                        if value == 'sphere':
                                # TODO: check if this is always suitable
                                vradius = 1/scene.range.x
                                self.vpoint2d = sphere(pos=vpos, radius=vradius,
 color=vcolor)
                        else:
                                vlabel = self._label or 'X'
                                self.vpoint2d = label(pos=vpos, text=vlabel, col
or=vcolor)
                                self.vpoint2d.linecolor = vcolor

                return locals()

        repr = property(**representation())

        def label():
                doc = "Label of VPoint2d"

                def fget(self):
                        return self._label

                def fset(self, value):
                        self._label = value
                        self.repr = 'label'

                return locals()

        label = property(**label())

        def position():
                doc = "Position of VPoint2"

                def fget(self):
                        if self.vpoint2d is not None:
                                return self.vpoint2d.pos
                        else:
                                return None

                def fset(self, posval):
                        self.point2d = Point_2(posval[0], posval[1])
                        if self.vpoint2d is not None:
                                self.vpoint2d.visible = False
                                if self.vpoint2d.__class__ is sphere:
                                        self.repr = 'sphere'
                                else:
                                        self.repr = 'label'

                return locals()

        pos = property(**position())

        def color():
                doc = "Color of VPoint2"
```

```
                def fget(self):
                        return self._color
                        #if self.vpoint2d is not None:
                        #       return self.vpoint2d.color
                        #else:
                        #       return None

                def fset(self, value):
                        self._color = value
                        if self.vpoint2d is not None:
                                self.vpoint2d.color = value
                                if self.vpoint2d.__class__ is label:
                                        self.vpoint2d.linecolor = value

                return locals()

        color = property(**color())

        def move(self):
                if self.vpoint2d.__class__ is label:
                        self.repr = 'sphere'
                pick = None
                while 1:
                        if scene.mouse.events:
                                event = scene.mouse.getevent()
                                if event.drag and event.pick is self.vpoint2d:
                                        dragpos = event.pickpos
                                        pick = event.pick
                                        #scene.cursor.visible = 0
                                elif event.drop:
                                        pick = None
                                        self.pos = scene.mouse.pos
                                        #scene.cursor.visible = 1
                                        break
                                if pick:
                                        newpos = scene.mouse.pos
                                        if newpos <> dragpos:
                                                pick.pos += newpos - dragpos
                                                dragpos = newpos

def worldSpacePos(frame, local):
        """Returns the position of local in world space."""
        xAxis = norm(frame.axis)
        zAxis = norm(cross(frame.axis, frame.up))
        yAxis = norm(cross(zAxis, xAxis))
        return frame.pos+local.x*xAxis+local.y*yAxis+local.z*zAxis

class VSegment2d(object):
        """
        Holds the visual representation of a CGAL.Segment_2() point ...
        """
        def __init__(self, segment2d=None):
                self.vsegment2d = None
                self.frame = frame()
                self.segment2d = segment2d

                self._color = None
                self._label = None

                if self.segment2d is not None:
                        # show the thing
                        pass
                else:
                        self.get()

        def get(self):
                self.vstart, self.vend = VPoint2d(), VPoint2d()
                self.vstart.vpoint2d.frame, self.vend.vpoint2d.frame = self.vseg
ment2d, self.vsegment2d
```



```
                self.segment2d = Segment_2(self.vstart.point2d, self.vend.point2
d)
                self.repr = 'curve'

        def representation():
                doc = "Representation of VSegment2d: visual.curve or visual.cylinder"

                def fget(self):
                        if self.vsegment2d.__class__ is curve:
                                return 'curve'
                        else:
                                return 'cylinder'

                def fset(self, value):
                        if value == 'curve':
                                self.lineseg = curve(pos=[self.vstart.pos, self.
vend.pos])
                                self.lineseg.frame = self.vsegment2d

                return locals()

        repr = property (**representation())


if __name__== "__main__":
        scene.range = 10.0

        #class mouseClick() tests:
        #while 1:
        #       a = mouseClick()
        #       print a
        #       a = mouseClick('left')
        #       print a
        #       a = mouseClick('right')
        #       print a
        #       a = mouseClick('middle')
        #       print a

        #while 1:
        #       a = VPoint2d()
        #       a.move()
        #
        #       a.repr = 'label'
        #       a.color = color.red
        #       scene.mouse.getclick()
        #       a.move()
        #       a.pos = (1,1)
        #       a = VPoint2d(point2d=Point_2(4,4), label='A')
        #       scene.mouse.getclick()
        #       a.pos = (-3,-2)

        a = VSegment2d()
```