Lab 9 Sensors I

1. Plane Sensor
The PlaneSensor node maps the mouse movement into the XY plane, moving the shape in the XY plane of its local coordinate system. This node allows you to limit the dragging operation to a rectangular area. In addition to the fields which are common to all dragging sensors, this node has the following fields:

- maxPosition which specifies the maximum X and Y.
- minPosition which specifies the minimum X and Y.

Note: if maxPosition is lower than minPosition for an axis, then the movement is not limited for that axis. By default, the movement is not limited in neither X or Y.

Syntax:
```markdown
PlaneSensor {
  enabled TRUE
  offset 0 0 0
  autoOffset TRUE
  maxPosition -1 -1
  minPosition 0 0
}
```

Exercise 1. Let’s create a cone that translates in the XY plane when the viewer drags it. Save it as “cone.wrl” and do “submit cmops080v-ap.f05 lab9 cone.wrl” on unix.ic.

```vrmpl
#VRML V2.0 utf8
Group {
  children [
    DEF OrangeCone Transform {
      children Shape {
        appearance Appearance {
          material Material {
            diffuseColor .52 .26 0
            specularColor .97 .98 .12
            emissiveColor .24 0 0
            ambientIntensity .00333
            shininess .12
            transparency .3
          }
        }
        geometry Cone {
          translation_changed TO OrangeCone.set_translation
        }
      }
    },
    DEF Sensor PlaneSensor {
  }
}
ROUTE Sensor.translation_changed TO OrangeCone.set_translation
```
2. **SphereSensor**
The SphereSensor node maps the mouse movement into a surface of a conceptual sphere, rotating the shape about the center of its local coordinate system.

Syntax:

```
SphereSensor {
    enabled TRUE
    offset 0 1 0 0
    autoOffset TRUE
}
```

**Exercise 2.** In the following VRML world we have a box that rotates when the viewer drags it. Let’s add one more box within a Group node. You need to define a Transform and SphereSensor. You don’t have to submit this code, but please write the code on this worksheet.

```vrml
Group {
    children [
        Group {
            children [
                DEF Shape Transform {
                    children Shape {
                        appearance Appearance {
                            material Material {}
                        }
                        geometry Box {}
                    },
                    DEF ShapeSensor SphereSensor {}
                },
                # create a group node for a box
            }
        }
        ROUTE ShapeSensor.rotation_changed TO Shape.set_rotation
        # create a ROUTE for the second box
```