X3D Python Language Binding

SIGGRAPH Web3D Standarization Meeting (Korea Chapter)

Los Angeles, USA

July 29, 2019

Myeong Won Lee

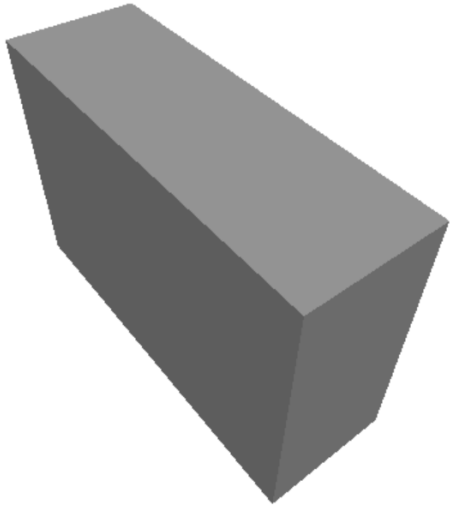
Status of Work

- ISO/IEC 19777-6 X3D Language Binding: Python
  - NWIP to be prepared

X3D Python Language Binding Concepts

- What is Python language binding?
  - X3D scene access interface using Python language
    - Specify 19775-2 X3D Scene Access Interface using Python language
  - Development of Python programs using X3D data types and functions
  - X3D scene read, update, store, and exchange in Python applications
- Scope
  - Provides a browser implementation independent way of accessing a browser's capabilities via the languages
  - Provides a set of implementation independent base classes and interfaces that represent possible interactions with an X3D scene through an SAI
  - Provides a Python API format for X3D scene access

A Simple Example of X3D Scene Access API



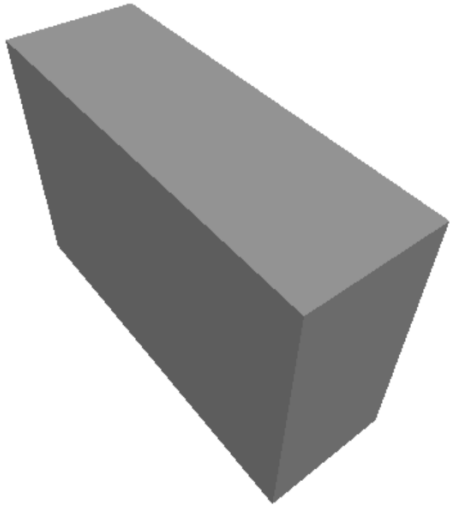
```
<X3D>
<Scene>
   <Background skyColor='1 1 1'/>
   <Viewpoint description='Book View'
orientation='-0.747 -0.624 -0.231 1.05' position='-
1.81 3.12 2.59'/>
   <Shape>
      <Box size='1 2 3'/>
      <Appearance>
         <Material/>
      </Appearance>
   </Shape>
</Scene>
</X3D>
```

X3D

getX3D	setX3D
getScene	setScene
getBackground	setBackground
getViewpoint	setViewpoint
getShape	setShape
getBox	setBox
getApperance	setApperance
getMaterial	setMaterial

X3D Scene Access Interface (SAI)

A Sample of X3D Scene Access API (C++)



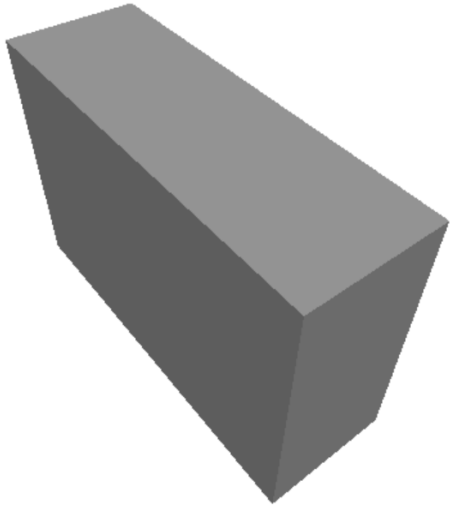
getX3D	setX3D
getScene	setScene
getBackground	setBackground
getViewpoint	setViewpoint
getShape	setShape
getBox	setBox
getApperance	setApperance
getMaterial	setMaterial

X3D Scene Access Interface (SAI)

getX3D (&pX3D) getScene(&pScene) getBackground(&pBackground) getViewpoint(&pViewpoint) getShape(&pShape) getBox(&pBox) getApperance(&pAppearance) getMaterial(&pMaterial)
setX3D (pX3D) setScene(pScene) setBackground(pBackground) setViewpoint(pViewpoint) setShape(pShape) setBox(pBox) setApperance(pAppearance) setMaterial(pMaterial)

X3D C++ encoding

A Sample of X3D Scene Access API (C#)



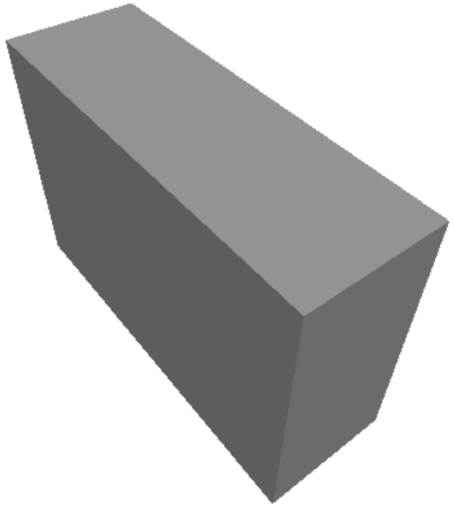
getX3D	setX3D
getScene	setScene
getBackground	setBackground
getViewpoint	setViewpoint
getShape	setShape
getBox	setBox
getApperance	setApperance
getMaterial	setMaterial

X3D Scene Access Interface (SAI)

getX3D (pX3D) getScene(pScene) getBackground(pBackground) getViewpoint(pViewpoint) getShape(pShape) getBox(pBox) getApperance(pAppearance) getMaterial(pMaterial)
setX3D (pX3D) setScene(pScene) setBackground(pBackground) setViewpoint(pViewpoint) setShape(pShape) setBox(pBox) setApperance(pAppearance) setMaterial(pMaterial)

X3D C# encoding

A Sample of X3D Scene Access API (Python)



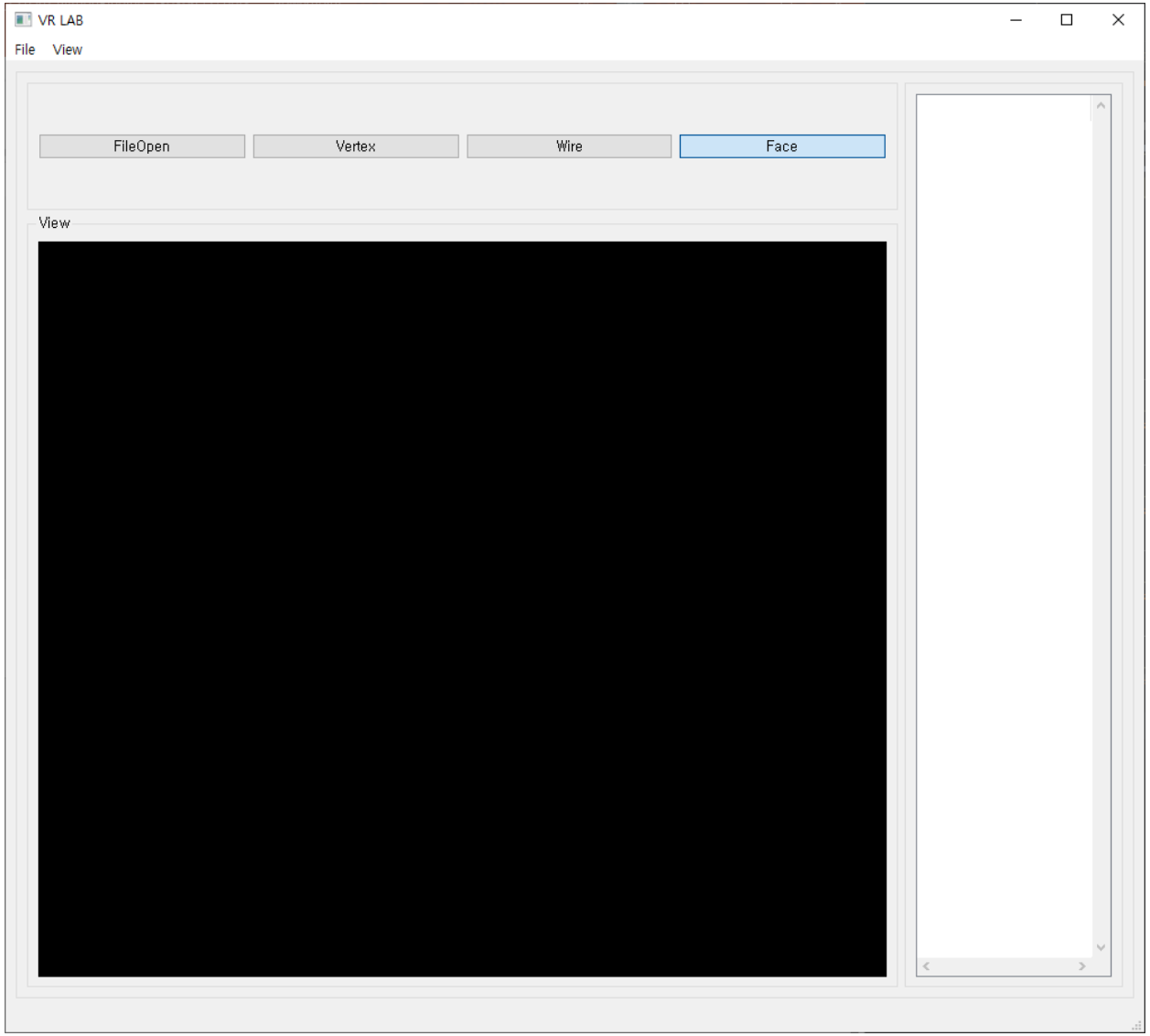
getX3D	setX3D
getScene	setScene
getBackground	setBackground
getViewpoint	setViewpoint
getShape	setShape
getBox	setBox
getApperance	setApperance
getMaterial	setMaterial

X3D Scene Access Interface (SAI)

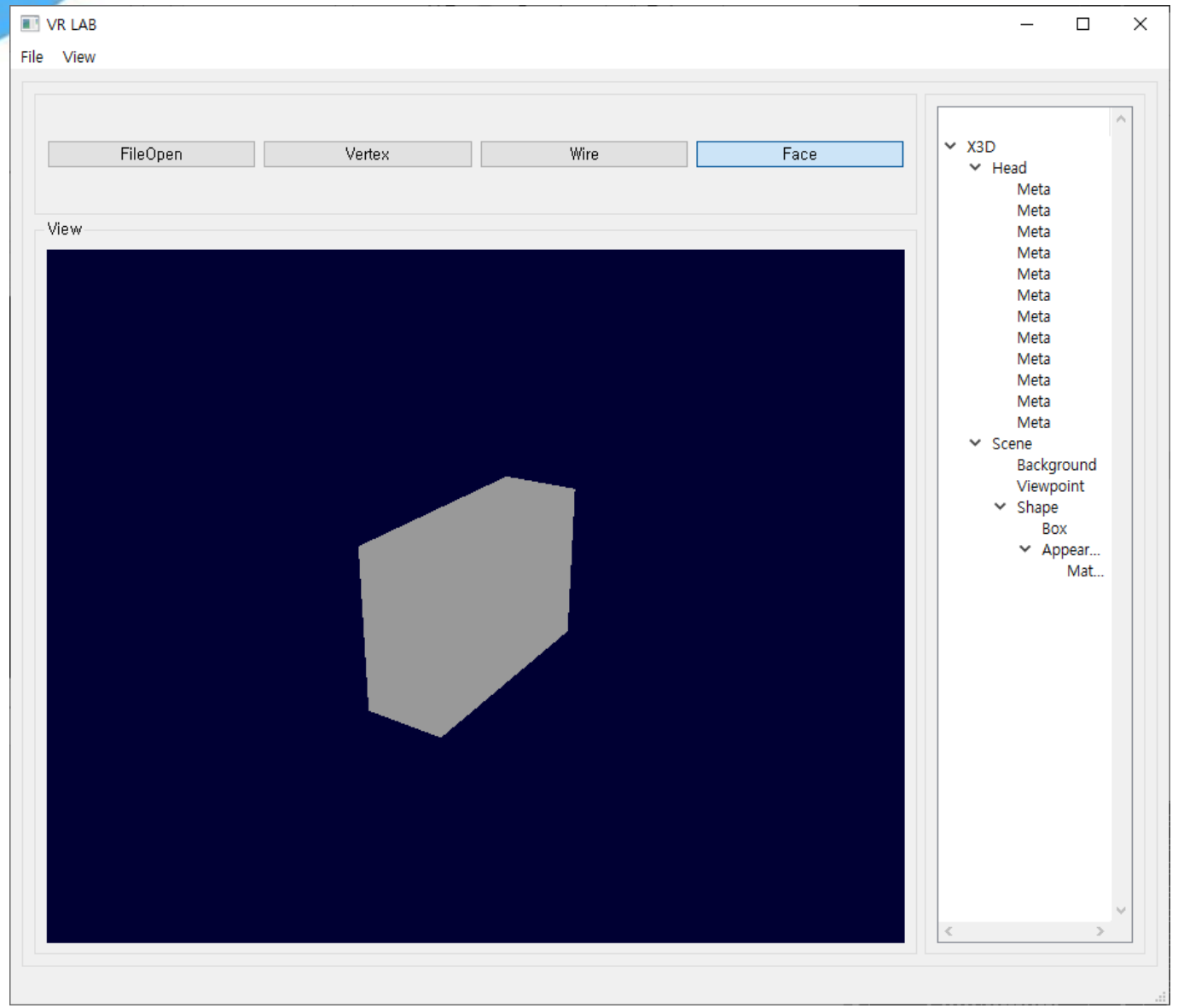
getX3D (pX3D) getScene(pScene) getBackground(pBackground) getViewpoint(pViewpoint) getShape(pShape) getBox(pBox) getApperance(pAppearance) getMaterial(pMaterial)
setX3D (pX3D) setScene(pScene) setBackground(pBackground) setViewpoint(pViewpoint) setShape(pShape) setBox(pBox) setApperance(pAppearance) setMaterial(pMaterial)

X3D Python encoding

Python Language Binding Viewer UI



Box.x3d


VR LAB
File
View
FileOpen
Vertex
Wire
Face
View
X3D
Head
Meta
Meta
Meta
Meta
Meta
Meta
Meta
Meta
Meta
Meta
Meta
Meta
Scene
Background
Viewpoint
Shape
Box
Appear...
Mat...

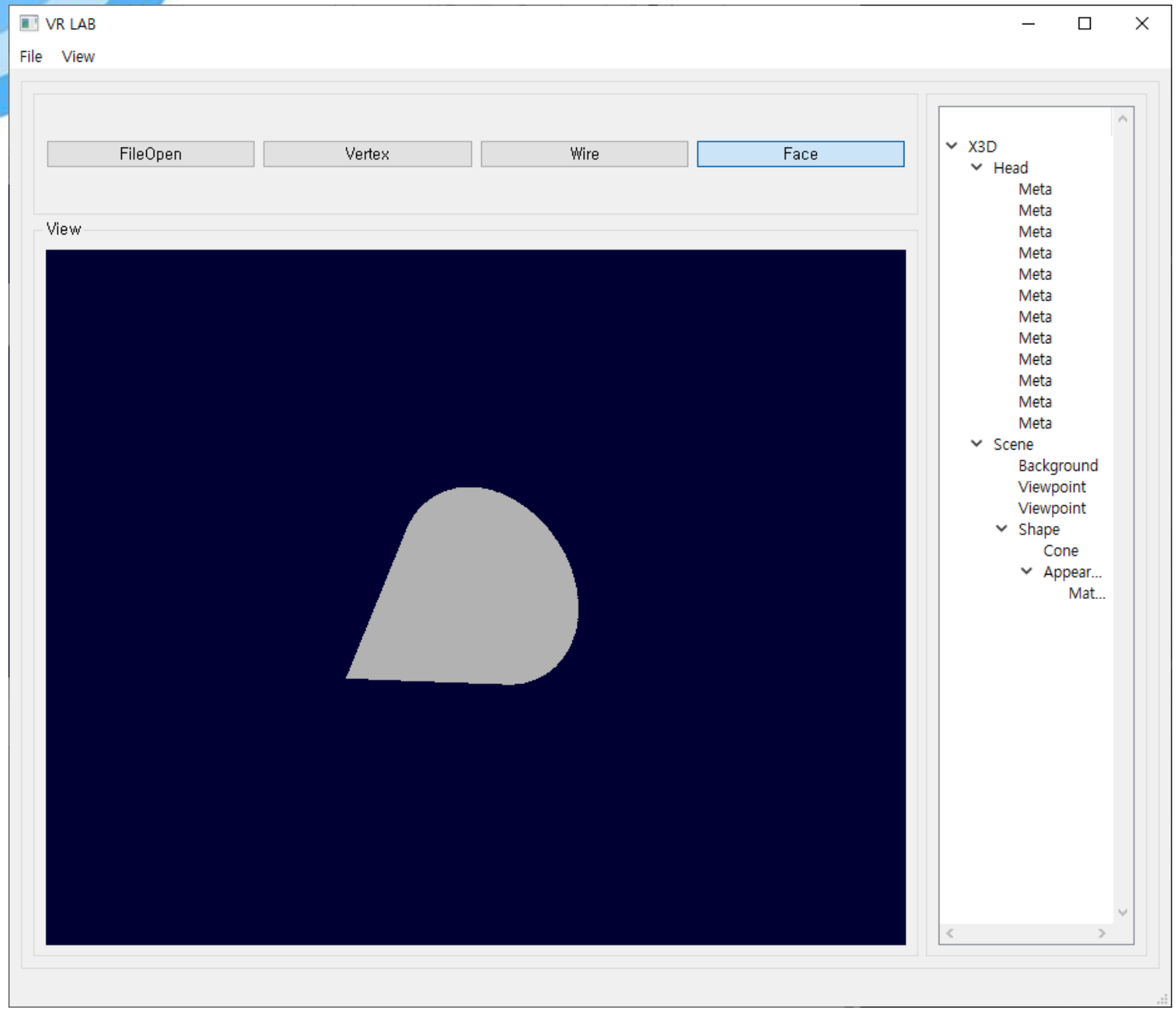
Box.x3d

```
15
16      def Draw(self,x,y,z):
17
18          self.xsize = x
19          self.ysize = y
20          self.zsize = z
21
22          point1 = [self.xsize / 2.0, self.ysize / 2.0, self.zsize / -2.0]
23          point2 = [self.xsize / 2.0, self.ysize / 2.0, self.zsize / 2.0]
24          point3 = [self.xsize / 2.0, self.ysize / -2.0, self.zsize / 2.0]
25          point4 = [self.xsize / 2.0, self.ysize / -2.0, self.zsize / -2.0]
26          point5 = [self.xsize / -2.0, self.ysize / -2.0, self.zsize / 2.0]
27          point6 = [self.xsize / -2.0, self.ysize / 2.0, self.zsize / 2.0]
28          point7 = [self.xsize / -2.0, self.ysize / 2.0, self.zsize / -2.0]
29          point8 = [self.xsize / -2.0, self.ysize / -2.0, self.zsize / -2.0]
30
```

Box.x3d

```
30
31      glBegin(GL_QUADS)
32
33      glVertex3fv(point1)
34      glVertex3fv(point2)
35      glVertex3fv(point6)
36      glVertex3fv(point7)
37
38      glVertex3fv(point3)
39      glVertex3fv(point4)
40      glVertex3fv(point8)
41      glVertex3fv(point5)
42
43      glVertex3fv(point2)
44      glVertex3fv(point3)
45      glVertex3fv(point5)
46      glVertex3fv(point6)
47
48      glVertex3fv(point7)
49      glVertex3fv(point8)
50      glVertex3fv(point4)
51      glVertex3fv(point1)
52
53      glVertex3fv(point6)
54      glVertex3fv(point5)
55      glVertex3fv(point8)
56      glVertex3fv(point7)
57
58      glVertex3fv(point1)
59      glVertex3fv(point4)
60      glVertex3fv(point3)
61      glVertex3fv(point2)
62
63      glEnd()
64
65
```

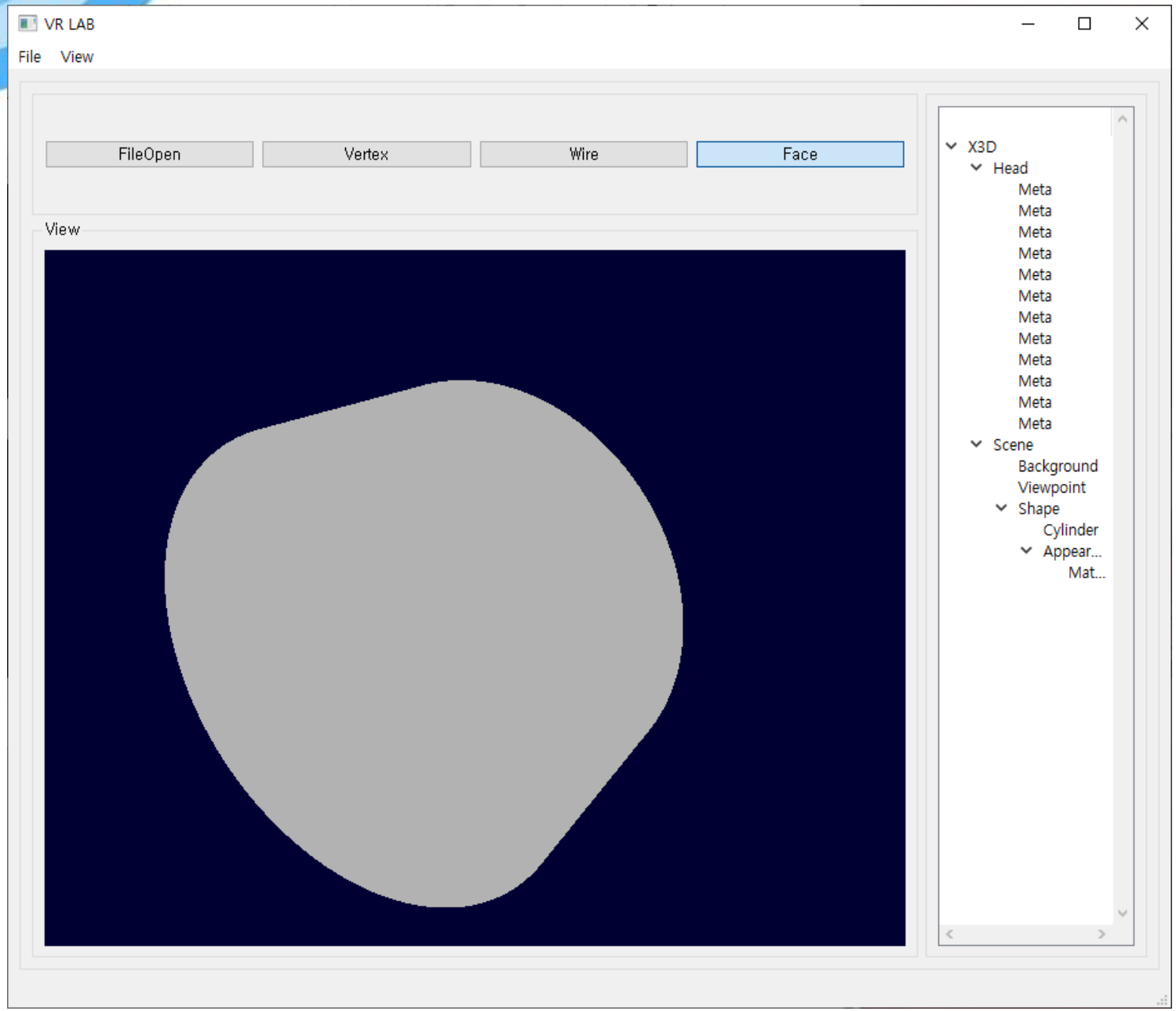
Cone.x3d


VR LAB
File
View
FileOpen
Vertex
Wire
Face
View
X3D
Head
Meta
Meta
Meta
Meta
Meta
Meta
Meta
Meta
Meta
Meta
Meta
Meta
Scene
Background
Viewpoint
Viewpoint
Shape
Cone
Appear...
Mat...

Cone.x3d

```
111     def Draw(self, r, h, bottom, side):
112
113         ui = Ui_MainWindow()
114
115         self.radius = r
116         self.height = h
117
118
119         glBegin(GL_TRIANGLE_FAN)
120         glVertex3f(0, 0, self.height)
121         angle = 0.0
122
123         for angle in range (0, 360) :
124             glVertex3f(sin(angle) * self.radius / 2, cos(angle) * self.radius / 2, 0)
125         glEnd()
126
127         glBegin(GL_TRIANGLE_FAN)
128         glVertex3f(0,0,0)
129         angle = 0.0
130         for angle in range(0, 360) :
131             glNormal3f(0, -1, 0)
132             glVertex3f(sin(angle) * self.radius / 2, cos(angle) * self.radius / 2, 0)
133         glEnd()
134
135
```

Cylinder.x3d



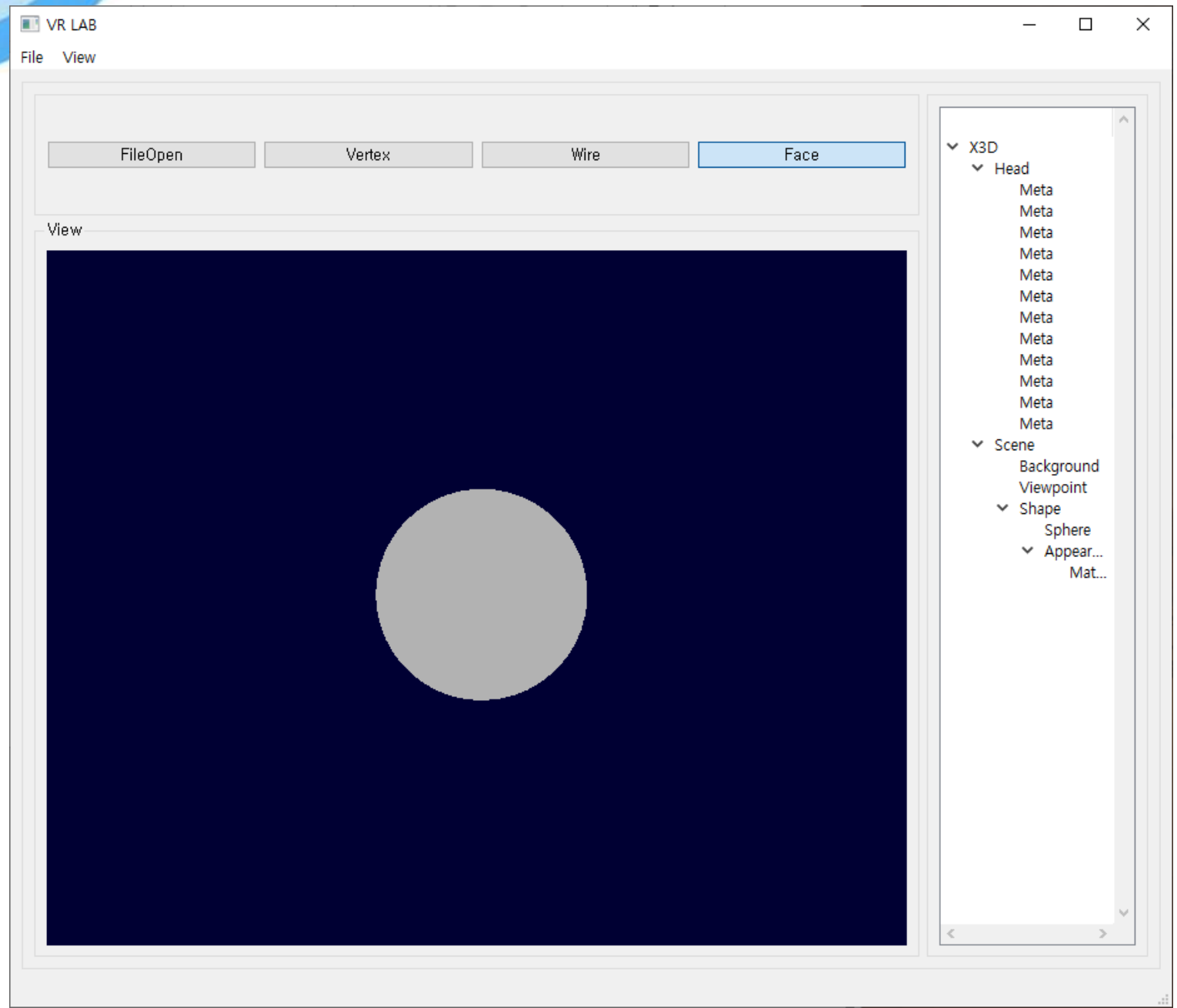
Cylinder.x3d

```
192     def Draw(self, r, h, top, bottom, side):
193
194         ui = Ui_MainWindow()
195
196         self.radius = r
197         self.height = h
198
199         x = 0.0
200         y = 0.0
201         angle = 0.0
202         angle_stepsize = 0.01
203         CONST_PI = 3.14159265
204
205         glBegin(GL_QUAD_STRIP)
206         for angle in self.myrange(0.0, 2 * CONST_PI, angle_stepsize) :
207             x = self.radius * cos(angle)
208             y = self.radius * sin(angle)
209             glVertex3f(x, y, self.height)
210             glVertex3f(x, y, 0.0)
211
212         glVertex3f(self.radius, 0.0, self.height)
213         glVertex3f(self.radius, 0.0, 0.0)
214         glEnd()
215
```

Cylinder.x3d

```
215
216         glBegin(GL_POLYGON)
217         angle = 0.0
218         for angle in self.myrange(0.0, 2 * CONST_PI, angle_stepsize) :
219             x = self.radius * cos(angle)
220             y = self.radius * sin(angle)
221             glVertex3f(x, y, self.height)
222
223         glVertex3f(self.radius, 0.0, self.height)
224         glEnd()
225
226         glBegin(GL_POLYGON)
227         angle = 0.0
228         for angle in self.myrange(0.0, 2 * CONST_PI, angle_stepsize) :
229             x = self.radius * cos(angle)
230             y = self.radius * sin(angle)
231             glVertex3f(x, y, 0)
232
233         glVertex3f(self.radius, 0.0, 0)
234         glEnd()
235
236
```

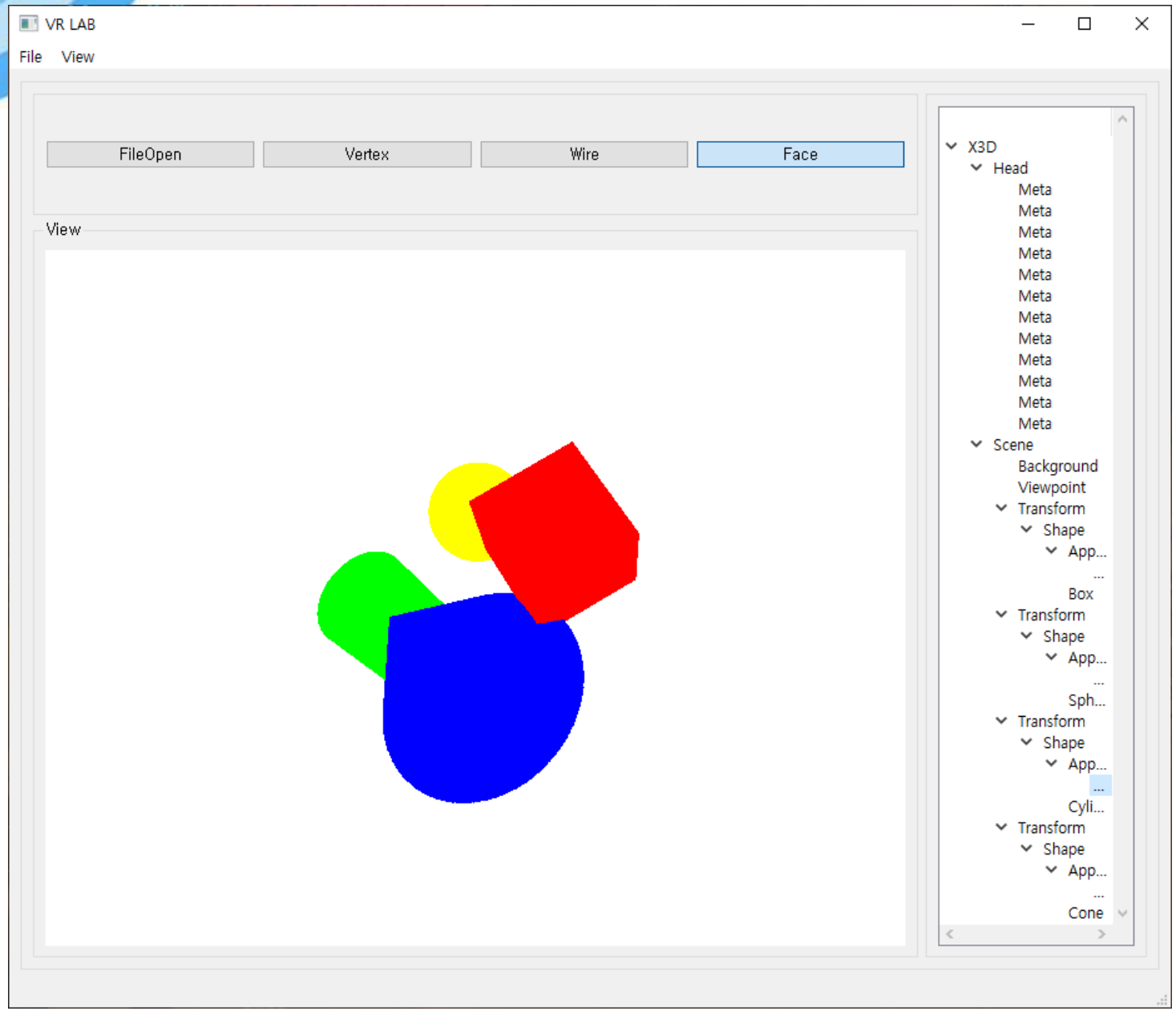
Sphere.x3d


VR LAB
File
View
FileOpen
Vertex
Wire
Face
View
X3D
Head
Meta
Meta
Meta
Meta
Meta
Meta
Meta
Meta
Meta
Meta
Meta
Meta
Scene
Background
Viewpoint
Shape
Sphere
Appear...
Mat...

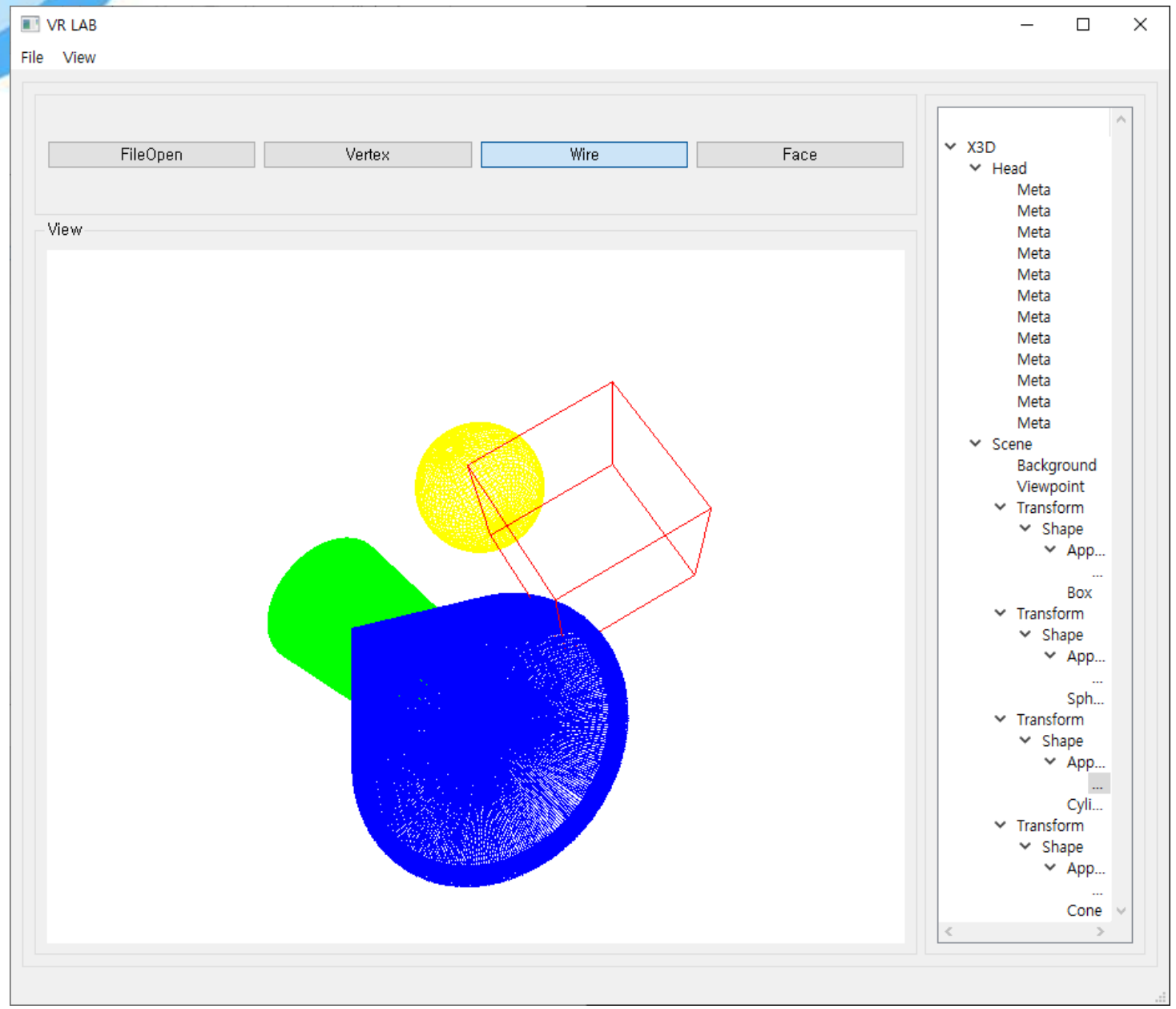
Sphere.x3d

```
302
303     def Draw(self, r):
304
305         ui = Ui_MainWindow
306
307         CONST_PI = 3.14159265
308         lats = 50
309         longs = 50
310
311         for i in range (0, lats) :
312             lat0 = float( CONST_PI * ( -0.5 + float (i - 1 ) / lats ) )
313             z0 = float( sin(lat0) )
314             zr0 = float( cos(lat0) )
315
316             lat1 = float ( CONST_PI * ( -0.5 + float (i) / lats ) )
317             z1 = float( sin(lat1) )
318             zr1 = float( cos(lat1) )
319
320             glBegin(GL_QUAD_STRIP)
321
322             for j in range (0, longs) :
323                 lng = 2 * CONST_PI * float( j - 1 ) / longs
324                 x = float ( cos(lng) )
325                 y = float ( sin(lng) )
326
327                 glNormal3f(x * zr0, y * zr0, z0)
328                 glVertex3f(x * zr0, y * zr0, z0)
329                 glNormal3f(x * zr1, y * zr1, z1)
330                 glVertex3f(x * zr1, y * zr1, z1)
331
332             glEnd()
333
```

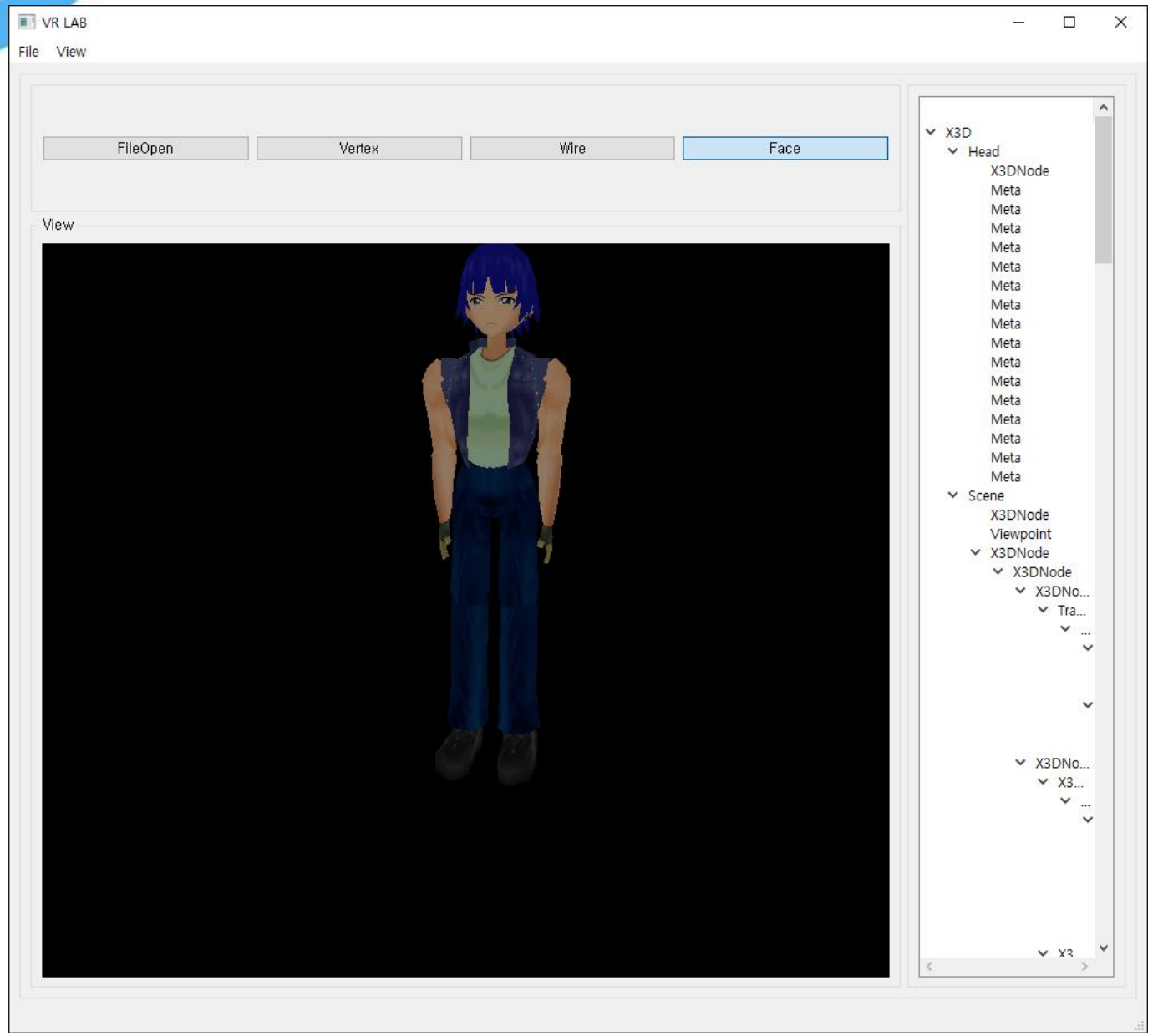
Transform.x3d



Transform.x3d



KoreanCharacter02Chul.x3d



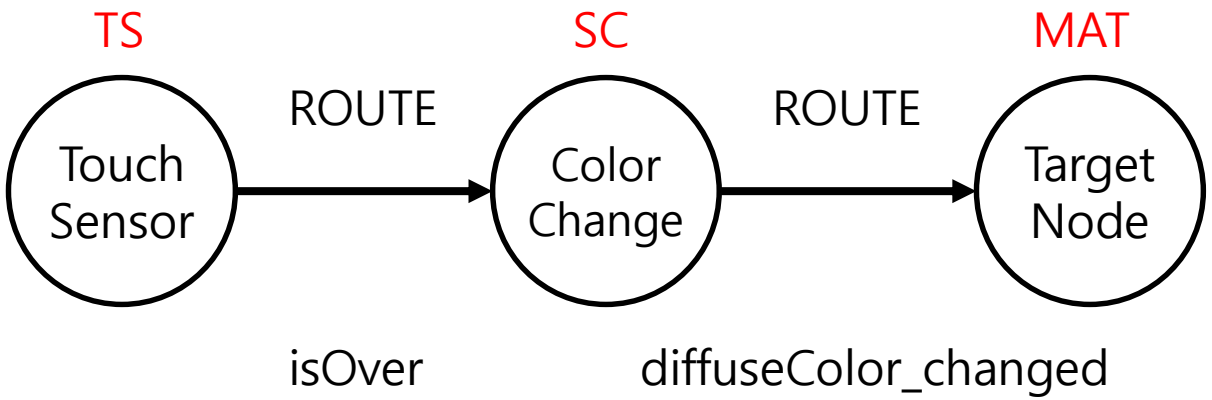
ISO/IEC NP 19777-6
X3D Python Language Binding
Annex C Examples
(Implementation)

Example 1. TouchSensor isOver event

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN" "http://www.web3d.org/specifications/x3d-3.0.dtd">
<X3D profile="Immersive">
  <head>
    <meta content="TouchSensorIsOverEvent.x3d" name="filename"/>
    <meta content="Xeena VRML importer" name="translator"/>
    <meta content="23 February 2005" name="imported"/>
    <meta content="23 February 2005" name="revised"/>
    <meta
content="X3D-Edit, http://www.web3D.org/TaskGroups/x3d/translation/README.X3D-Edit.html" name="generator"/>
    <meta content="Vrml97ToX3dNist, http://ovrt.nist.gov/v2_x3d.html" name="generator"/>
  </head>
  <Scene>
    <Group>
      <Shape>
        <Appearance>
          <Material DEF="MAT" diffuseColor="0 0 1"/>
        </Appearance>
        <Box/>
      </Shape>
      <TouchSensor DEF="TS"/>
    </Group>
    <Script DEF="SC" url="&quot;SAIExample1.class&quot; ">
      <field accessType="inputOnly" name="isOver" type="SFBool"/>
      <field accessType="outputOnly" name="diffuseColor_changed" type="SFColor"/>
    </Script>
    <ROUTE fromField="isOver" fromNode="TS" toField="isOver" toNode="SC"/>
    <ROUTE fromField="diffuseColor_changed" fromNode="SC"
      toField="set_diffuseColor" toNode="MAT"/>
```

Example 1. TouchSensor isOver event (Implementation Overview)

```
<Material DEF="MAT" diffuseColor="0 0 1"/>
<TouchSensor DEF="TS"/>
<Script DEF="SC" url="&quot;SAIExample1.class&quot; ">
    <field accessType="inputOnly" name="isOver" type="SFBool"/>
    <field accessType="outputOnly" name="diffuseColor_changed" type="SFColor"/>
</Script>
<ROUTE fromField="isOver" fromNode="TS" toField="isOver" toNode="SC"/>
<ROUTE fromField="diffuseColor_changed" fromNode="SC" toField="set_diffuseColor" tooNode="MAT"/>
```



Example 1. TouchSensor (Implementation) (1)

```
from X3DLib import *
import sys, os

from PyQt5.QtWidgets import *
from PyQt5.QtGui import *
from PyQt5.QtCore import *
from PyQt5.QtWidgets import QApplication, QMainWindow, QWidget, QOpenGLWidget
import platform


class SAIExample(QOpenGLWidget):
    m_pScene = CX3DScene()

    lis = CX3DFieldEventListener()

    RED = [1.0, 0.0, 0.0]
    BLUE = [0.0, 0.0, 1.0]

    def  __init__(self, parent = None):
        super(SAIExample, self).__init__(parent)

    def initializeGL(self):
        path = str(os.getcwd()) + "/2.x3d"
        path = path.replace('₩₩','/')
        self.m_pScene.Parsing(path)

        glPolygonMode(GL_FRONT, GL_FILL)
        glPolygonMode(GL_BACK, GL_FILL)
```

Example 1. TouchSensor (Implementation) (2)

```
        glShadeModel(GL_SMOOTH)
        glEnable(GL_NORMALIZE)

        glClearColor(0.0, 0.0, 0.0, 1.0)

        glClearDepth(1.0)
        glPolygonMode(GL_FRONT_AND_BACK, GL_FILL)

        glEnable(GL_DEPTH_TEST)
        glEnable(GL_TEXTURE_2D)

        self.isOver = CSFBool(self.m_pScene.m_fields.get("isOver"))
        self.diffuseColor = CSFColor(self.m_pScene.m_fields.get("diffuseColor_changed"))

        self.m_pScene.m_TouchSensor.setField(self.isOver)
        self.m_pScene.m_Script.setField(self.diffuseColor)

        self.isOver.addX3DEventListner(self.lis)
def resizeGL(self, width, height):
        glGetError()

        aspect = width if (height == 0) else width / height

        glViewport(0, 0, width, height)
        glMatrixMode(GL_PROJECTION)
        glLoadIdentity()
        gluPerspective(45, aspect, 0.1, 1000.0)
        glMatrixMode(GL_MODELVIEW)
        glLoadIdentity()
```

Example 1. TouchSensor (Implementation) (3)

```
    def paintGL(self):
        glPolygonMode(GL_FRONT, GL_FILL)
        glPolygonMode(GL_BACK, GL_FILL)

        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
        glLoadIdentity()
        glTranslatef(0.0,0.0,-5.0)
        glRotatef(60.0, 1.0, 1.0, 0.0)
        self.m_pScene.Draw()
        glFlush()

        self.update()

    def mousePressEvent(self, event):
        event = self.m_pScene.m_TouchSensor
        self.isOver.setValue(True)
        self.readableFieldChanged(event)
        self.m_pScene.m_Script.setDiffuseColor(self.diffuseColor)
        self.m_pScene.Draw()
        self.update()

def mouseReleaseEvent(self, event):
        event = self.m_pScene.m_TouchSensor
        self.isOver.setValue(False)
        self.readableFieldChanged(event)
        self.m_pScene.m_Script.setDiffuseColor(self.diffuseColor)
        self.m_pScene.Draw()
        self.update()
```

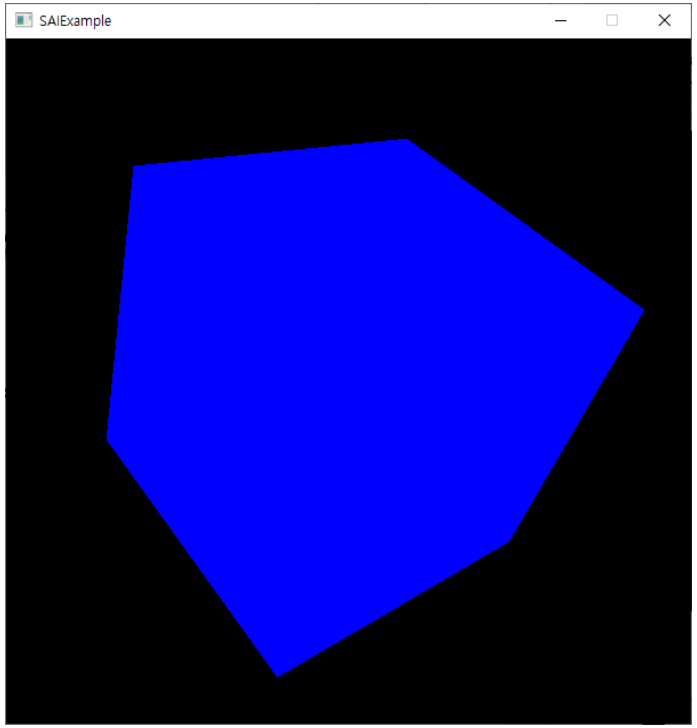
Example 1. TouchSensor (Implementation) (4)

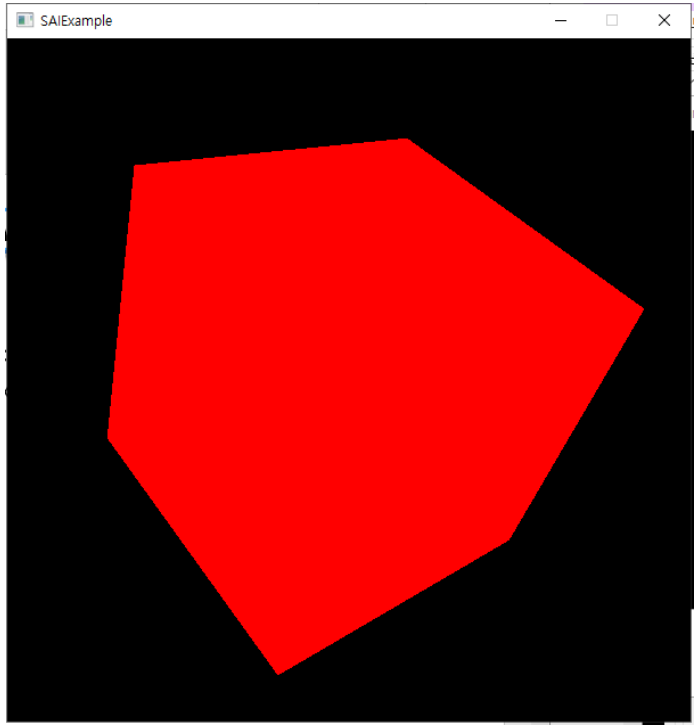
```
 def readableFieldChanged(self, evt):
        if evt.getSource() == self.isOver:
            if self.isOver.getValue() == True:
                self.diffuseColor.setValue1(SAIExample.RED)
            else:
                self.diffuseColor.setValue1(SAIExample.BLUE)
        else:
            print("Unhandled event : ", evt)

if __name__=='__main__':
    app = QApplication(sys.argv)
    window = SAIExample()
    window.setWindowTitle('SAIExample')
    window.setFixedSize(600,600)
    window.show()
    sys.exit(app.exec_())
```

Example 1. TouchSensor isOver event.x3d (Implementation) (5)

On click, an event is generated

- On click, the color of the box is changed from blue to red.
- On click in the outside of the box, the color of the box is changed to blue.





Example 2. Create Nodes (Implementation) (1)

```
from X3DLib import *
import sys, os

from PyQt5.QtWidgets import *
from PyQt5.QtGui import *
from PyQt5.QtCore import *
from PyQt5.QtWidgets import QApplication, QMainWindow, QWidget, QOpenGLWidget
import platform


class SAIExample(QOpenGLWidget):
    m_pScene = CX3DScene()

    def  __init__(self, parent = None):
        super(SAIExample, self).__init__(parent)

    def initializeGL(self):
        path = str(os.getcwd()) + "/3.x3d"
        path = path.replace('₩₩','/')
        self.m_pScene.Parsing(path)

        glPolygonMode(GL_FRONT, GL_FILL)
        glPolygonMode(GL_BACK, GL_FILL)
```

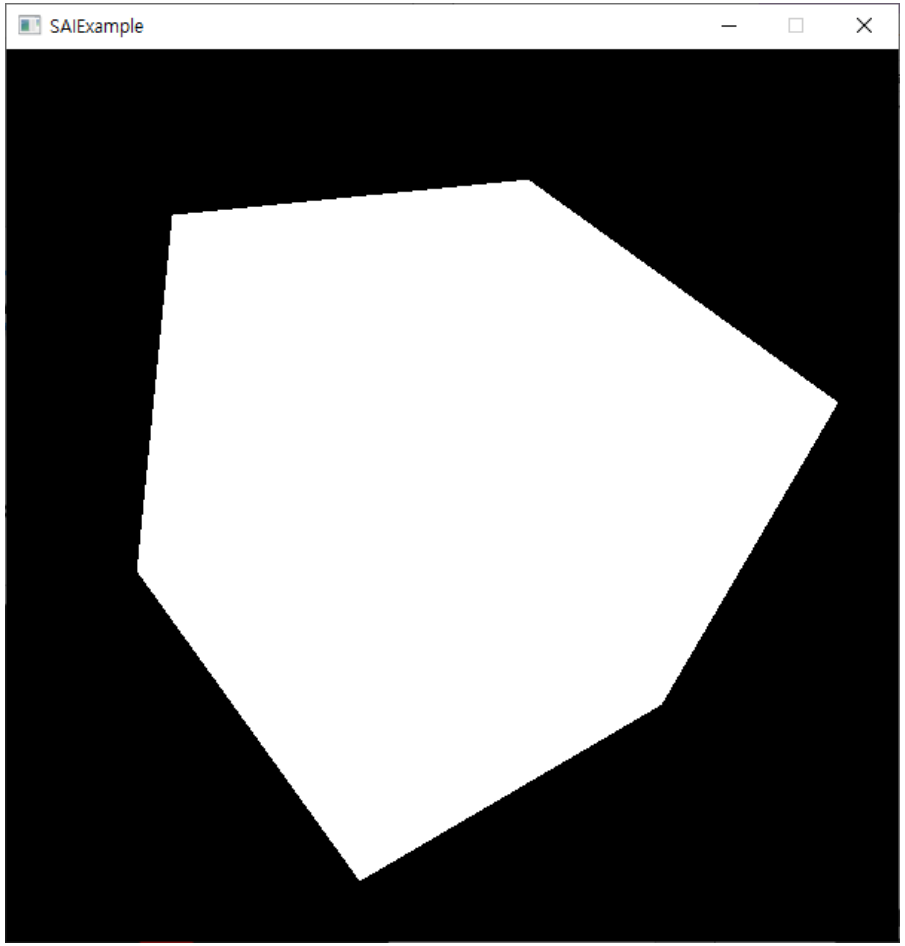
Example 2. Create Nodes (Implementation) (2)

```
glShadeModel(GL_SMOOTH)
        glEnable(GL_NORMALIZE)

        glClearColor(0.0, 0.0, 0.0, 1.0)

        glClearDepth(1.0)
        glPolygonMode(GL_FRONT_AND_BACK, GL_FILL)

        glEnable(GL_DEPTH_TEST)
        glEnable(GL_TEXTURE_2D)

        self.shape = CShape(self.m_pScene.createNode("Shape"))
        self.box = CBox(self.m_pScene.createNode("Box"))

        self.shape.setGeometry(self.box)
        self.m_pScene.m_Scene.addRootNode(self.shape)

    def resizeGL(self, width, height):
        glGetError()

        aspect = width if (height == 0) else width / height

        glViewport(0, 0, width, height)
        glMatrixMode(GL_PROJECTION)
        glLoadIdentity()
        gluPerspective(45, aspect, 0.1, 1000.0)
        glMatrixMode(GL_MODELVIEW)
        glLoadIdentity()
```

Example 2. Create Nodes (Implementation) (3)

```
 def paintGL(self):
        glPolygonMode(GL_FRONT, GL_FILL)
        glPolygonMode(GL_BACK, GL_FILL)

        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
        glLoadIdentity()
        glTranslatef(0.0,0.0,-5.0)
        glRotatef(60.0, 1.0, 1.0, 0.0)
        self.m_pScene.Draw()
        glFlush()

        self.update()

if __name__=='__main__':
    app = QApplication(sys.argv)
    window = SAIExample()
    window.setWindowTitle('SAIExample')
    window.setFixedSize(600,600)
    window.show()
    sys.exit(app.exec_())
```

Example 2. Create Nodes (Implementation) (4)



Example 3. Per frame notification (Implementation) (1)

```
from X3DLib import *
import sys, os

from PyQt5.QtWidgets import *
from PyQt5.QtGui import *
from PyQt5.QtCore import *
from PyQt5.QtWidgets import QApplication, QMainWindow, QWidget, QOpenGLWidget
import platform


class SAIExample(QOpenGLWidget):
    m_pScene = CX3DScene()

    def  __init__(self, parent = None):
        super(SAIExample, self).__init__(parent)

    def initializeGL(self):
        path = str(os.getcwd()) + "/3.x3d"
        path = path.replace('₩₩','/')
        self.m_pScene.Parsing(path)

        glPolygonMode(GL_FRONT, GL_FILL)
        glPolygonMode(GL_BACK, GL_FILL)
```

Example 3. Per frame notification (Implementation) (2)

```
        glShadeModel(GL_SMOOTH)
        glEnable(GL_NORMALIZE)

        glClearColor(0.0, 0.0, 0.0, 1.0)

        glClearDepth(1.0)
        glPolygonMode(GL_FRONT_AND_BACK, GL_FILL)

        glEnable(GL_DEPTH_TEST)
        glEnable(GL_TEXTURE_2D)
        print(self.m_pScene.children)

    def resizeGL(self, width, height):
        glGetError()

        aspect = width if (height == 0) else width / height

        glViewport(0, 0, width, height)
        glMatrixMode(GL_PROJECTION)
        glLoadIdentity()
        gluPerspective(45, aspect, 0.1, 1000.0)
        glMatrixMode(GL_MODELVIEW)
        glLoadIdentity()
```

Example 3. Per frame notification (Implementation) (3)

```
def prepareEvents(self):
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

    self.frameTime = (int(round(time.time() * 1000)) - self.lastStartTime) / 1000.0
    self.lastStartTime = int(round(time.time() * 1000))

    if self.frameTime == 0:
        return

    fps = 1.0 / self.frameTime

    s = "FPS: " + str(fps)

    glColor3f(1.0, 1.0, 1.0)
    glPushMatrix()
    glRasterPos(0, 0)
    for ch in s :
        glutBitmapCharacter(GLUT_BITMAP_9_BY_15, ctypes.c_int(ord(ch)))

    glPopMatrix()
    glFlush()
```

Example 3. Per frame notification (Implementation) (4)

```
    def paintGL(self):
        glPolygonMode(GL_FRONT, GL_FILL)
        glPolygonMode(GL_BACK, GL_FILL)

        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

        self.prepareEvents()

        glLoadIdentity()
        glTranslatef(0.0,0.0,-5.0)
        glRotatef(60.0, 1.0, 1.0, 0.0)
        self.m_pScene.Draw()
        glFlush()

        self.update()

if __name__=='__main__':
    glutInit()
    app = QApplication(sys.argv)
    window = SAIExample()
    window.setWindowTitle('SAIExample')
    window.setFixedSize(600,600)
    window.show()
    sys.exit(app.exec_())
```

Example 3. Per frame notification (Implementation) (5)

prepareEvents()
FPS is calculated and displayed

Example 4. Add dynamic routes (Implementation) (1)

```
from X3DLib import *
import sys, os

from PyQt5.QtWidgets import *
from PyQt5.QtGui import *
from PyQt5.QtCore import *
from PyQt5.QtWidgets import QApplication, QMainWindow, QWidget, QOpenGLWidget
import platform


class SAIExample(QOpenGLWidget):
    m_pScene = CX3DScene()

    lis = CX3DFieldEventListener()

    def  __init__(self, parent = None):
        super(SAIExample, self).__init__(parent)

    def initializeGL(self):
        path = str(os.getcwd()) + "/5.x3d"
        path = path.replace('₩₩','/')
        self.m_pScene.Parsing(path)

        glPolygonMode(GL_FRONT, GL_FILL)
        glPolygonMode(GL_BACK, GL_FILL)

        glShadeModel(GL_SMOOTH)
        glEnable(GL_NORMALIZE)

        glClearColor(0.0, 0.0, 0.0, 1.0)

        glClearDepth(1.0)
        glPolygonMode(GL_FRONT_AND_BACK, GL_FILL)
```

Example 4. Add dynamic routes (Implementation) (2)

```
glEnable(GL_DEPTH_TEST)
        glEnable(GL_TEXTURE_2D)

        self.touchTime = CSFTime(self.m_pScene.m_fields.get("touchTime"))
        self.touchTime.addX3DEventListner(self.lis)

        self.shape = CShape(self.m_pScene.createNode("Shape"))
        self.box = CBox(self.m_pScene.createNode("Box"))
        self.touchSensor = self.m_pScene.createNode("TouchSensor")

        self.shape.setGeometry(self.box)

        self.group = CGroup(self.m_pScene.createNode("Group"))

        self.group.addChildren(self.shape)
        self.group.addChildren(self.touchSensor)

        self.m_pScene.m_Scene.addRootNode(self.group)

        self.selfRef = CX3DScriptNode(self.m_pScene.getNode("SC"))
        self.m_pScene.addRoute(self.touchSensor, "touchTime", self.selfRef, "touchTime")
        self.m_pScene.m_TouchSensor.setField(self.touchTime)

    def resizeGL(self, width, height):
        glGetError()

        aspect = width if (height == 0) else width / height

        glViewport(0, 0, width, height)
        glMatrixMode(GL_PROJECTION)
        glLoadIdentity()
        gluPerspective(45, aspect, 0.1, 1000.0)
        glMatrixMode(GL_MODELVIEW)
        glLoadIdentity()
```

Example 4. Add dynamic routes (Implementation) (3)

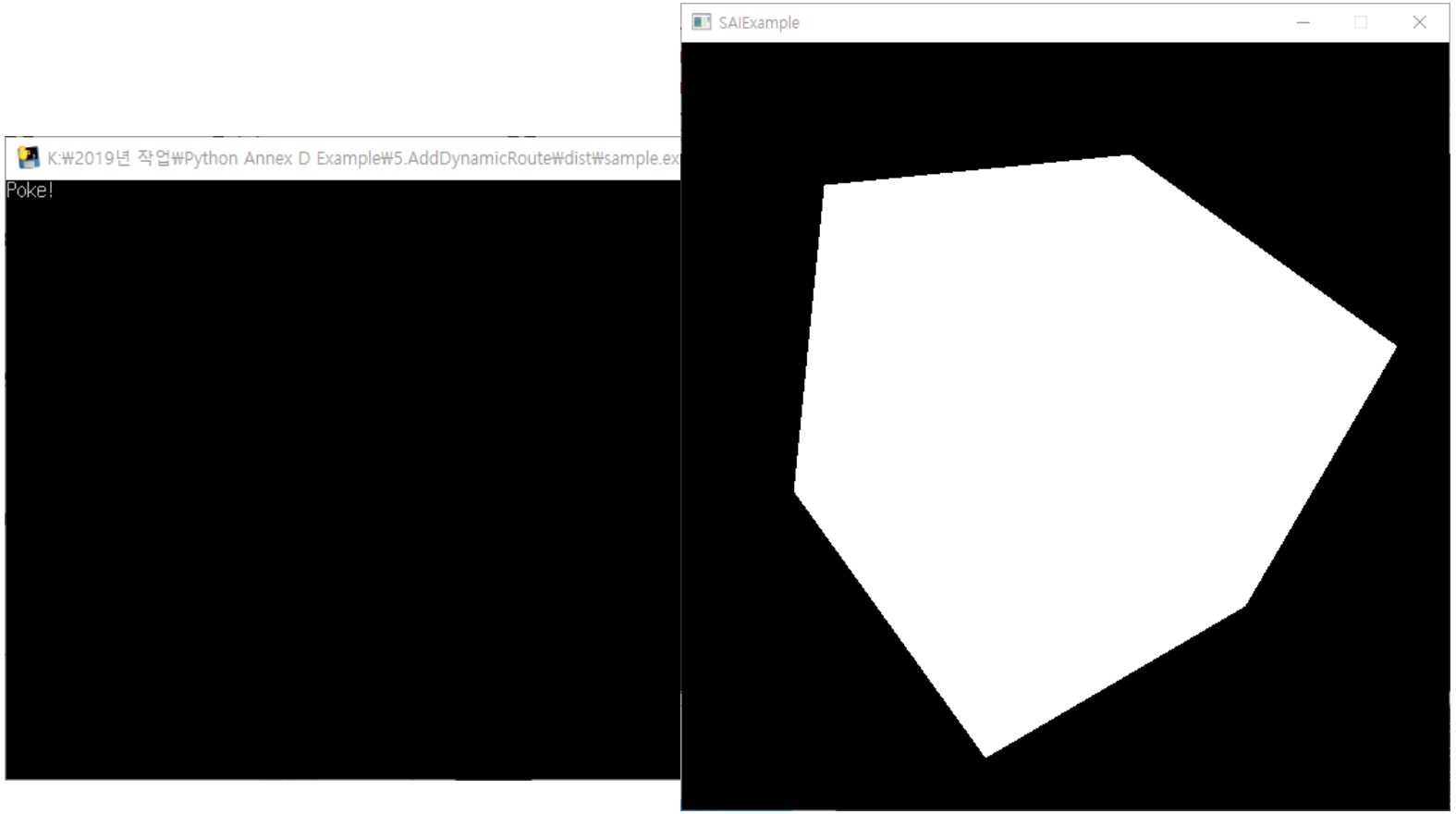
```
    def paintGL(self):
        glPolygonMode(GL_FRONT, GL_FILL)
        glPolygonMode(GL_BACK, GL_FILL)

        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
        glLoadIdentity()
        glTranslatef(0.0,0.0,-5.0)
        glRotatef(60.0, 1.0, 1.0, 0.0)
        self.m_pScene.Draw()
        glFlush()

        self.update()

    def mousePressEvent(self, event):
        event = self.m_pScene.m_TouchSensor
        self.readableFieldChanged(event)

    def readableFieldChanged(self, evt):
        if evt.getSource() == self.touchTime:
            print("Poke!")
        else:
            print("Unhandled event : ", evt)

if __name__=='__main__':
    app = QApplication(sys.argv)
    window = SAIExample()
    window.setWindowTitle('SAIExample')
    window.setFixedSize(600,600)
    window.show()
    sys.exit(app.exec_())
```

Example 4. Add dynamic routes (Implementation) (4)

In the initialization

Create Scene > Group > Shape > Box > TouchSensor

On clicking Box, “Poke!” is displayed



Work in Progress

- 19777-6 NWIP preparation
- Implementation of Python language bindings
  - 19777-6 X3D scene access interface definition using Python
    - Python and PyOpenGL
- Developing X3D Binding viewer programs with Python binding capability