Corrigés de travaux Dirigés

Partie 1 :

La méthode Display doit contenir par exemple le code suivant :

```
   public void display( GLAutoDrawable drawable )
{
      final GL2 gl = drawable.getGL().getGL2();

      //drawing top
      gl.glBegin ( GL2.GL_LINES );
      gl.glVertex3f( -0.3f, 0.3f, 0 );
      gl.glVertex3f( 0.3f,0.3f, 0 );
      gl.glEnd();

      //drawing bottom
      gl.glBegin( GL2.GL_LINES );
      gl.glVertex3f( -0.3f,-0.3f, 0 );
      gl.glVertex3f( 0.3f,-0.3f, 0 );
      gl.glEnd();

      //drawing the right edge
      gl.glBegin( GL2.GL_LINES );
      gl.glVertex3f( -0.3f,0.3f, 0 );
      gl.glVertex3f( -0.3f,-0.3f, 0 );
      gl.glEnd();

      //drawing the left edge
      gl.glBegin( GL2.GL_LINES );
      gl.glVertex3f( 0.3f,0.3f,0 );
      gl.glVertex3f( 0.3f,-0.3f,0 );
      gl.glEnd();

      //building roof
      //building lft dia
      gl.glBegin( GL2.GL_LINES );
      gl.glVertex3f( 0f,0.6f, 0 );
      gl.glVertex3f( -0.3f,0.3f, 0 );
      gl.glEnd();
      //building rt dia
      gl.glBegin( GL2.GL_LINES );
      gl.glVertex3f( 0f,0.6f, 0 );
      gl.glVertex3f( 0.3f,0.3f, 0 );
      gl.glEnd();

      //building door
      //drawing top
      gl.glBegin ( GL2.GL_LINES );
      gl.glVertex3f( -0.05f, 0.05f, 0 );
      gl.glVertex3f( 0.05f, 0.05f, 0 );
      gl.glEnd();

      //drawing the left edge
      gl.glBegin ( GL2.GL_LINES );
      gl.glVertex3f( -0.05f, 0.05f, 0 );
      gl.glVertex3f( -0.05f, -0.3f, 0 );
      gl.glEnd();

      //drawing the right edge
      gl.glBegin ( GL2.GL_LINES );
      gl.glVertex3f( 0.05f, 0.05f, 0 );
      gl.glVertex3f( 0.05f, -0.3f, 0 );
      gl.glEnd();
   }
```

Partie 2 :

Polygone rouge :

```
public void display( GLAutoDrawable drawable ) {

      final GL2 gl = drawable.getGL().getGL2();
      gl.glColor3f( 1f,0f,0f ); //applying red

      gl.glBegin( GL2.GL_POLYGON );

      gl.glVertex3f( 0f,0.5f,0f  );
      gl.glVertex3f( -0.5f,0.2f,0f );
      gl.glVertex3f( -0.5f,-0.2f,0f );
      gl.glVertex3f( 0f,-0.5f,0f );
      gl.glVertex3f( 0f,0.5f,0f );
      gl.glVertex3f( 0.5f,0.2f,0f );
      gl.glVertex3f( 0.5f,-0.2f,0f );
      gl.glVertex3f( 0f,-0.5f,0f );

      gl.glEnd();
   }
```

Triangle à 3 couleurs différentes aux sommets :

```
public void display( GLAutoDrawable drawable ) {

      final GL2 gl = drawable.getGL().getGL2();
      gl.glBegin( GL2.GL_TRIANGLES );

      // Drawing Using Triangles

      gl.glColor3f( 1.0f, 0.0f, 0.0f );    // Red
      gl.glVertex3f( 0.5f,0.7f,0.0f );     // Top

      gl.glColor3f( 0.0f,1.0f,0.0f );      // green
      gl.glVertex3f( -0.2f,-0.50f,0.0f ); //
Bottom Left

      gl.glColor3f( 0.0f,0.0f,1.0f );      // blue
      gl.glVertex3f( 0.5f,-0.5f,0.0f );    //
Bottom Right

      gl.glEnd();
   }
```

Partie 3 :

Triangle avec application d'un changement d'échelle :

```
public void display( GLAutoDrawable drawable ) {

      final GL2 gl = drawable.getGL().getGL2();
      gl.glScalef( 0.50f,0.25f,0.50f );
      gl.glBegin( GL2.GL_TRIANGLES );
```

```
      // Drawing Using Triangles
      gl.glColor3f( 1.0f, 0.0f, 0.0f );    // Red
      gl.glVertex3f( 0.5f,0.7f,0.0f );     // Top

      gl.glColor3f( 0.0f,1.0f,0.0f );      // blue
      gl.glVertex3f( -0.2f,-0.50f,0.0f ); //
Bottom Left

      gl.glColor3f( 0.0f,0.0f,1.0f );      // green
      gl.glVertex3f( 0.5f,-0.5f,0.0f );    //
Bottom Right

      gl.glEnd();
   }
```

Triangle avec l'application d'une rotation à une angle variable :

```
public void display( GLAutoDrawable drawable ) {

      final GL2 gl = drawable.getGL().getGL2();
      gl.glClear (GL2.GL_COLOR_BUFFER_BIT |
GL2.GL_DEPTH_BUFFER_BIT );

      // Clear The Screen And The Depth Buffer
      gl.glLoadIdentity();  // Reset The View

      //triangle rotation
      gl.glRotatef( rtri, 0.0f, 1.0f, 0.0f );

      // Drawing Using Triangles
      gl.glBegin( GL2.GL_TRIANGLES );

      gl.glColor3f( 1.0f, 0.0f, 0.0f );    // Red
      gl.glVertex3f( 0.5f,0.7f,0.0f );     // Top
      gl.glColor3f( 0.0f,1.0f,0.0f );      // blue
      gl.glVertex3f( -0.2f,-0.50f,0.0f ); //
Bottom Left
      gl.glColor3f( 0.0f,0.0f,1.0f );      // green
      gl.glVertex3f( 0.5f,-0.5f,0.0f );    //
Bottom Right
```

```
      gl.glEnd();
      gl.glFlush();

      rtri += 0.2f;  //assigning the angle
   }
```

Partie 4 :

Cube avec des facettes colorées :

```
   public void display( GLAutoDrawable drawable )
{

      final GL2 gl = drawable.getGL().getGL2();
      gl.glClear(GL2.GL_COLOR_BUFFER_BIT |
GL2.GL_DEPTH_BUFFER_BIT );
      gl.glLoadIdentity();
      gl.glTranslatef( 0f, 0f, -5.0f );

      // Rotate The Cube On X, Y & Z
      gl.glRotatef(rquad, 1.0f, 1.0f, 1.0f);

      //giving different colors to different sides
      gl.glBegin(GL2.GL_QUADS); // Start Drawing
The Cube
      gl.glColor3f(1f,0f,0f); //red color
      gl.glVertex3f(1.0f, 1.0f, -1.0f); // Top
Right Of The Quad (Top)
      gl.glVertex3f( -1.0f, 1.0f, -1.0f); // Top
Left Of The Quad (Top)
      gl.glVertex3f( -1.0f, 1.0f, 1.0f ); //
Bottom Left Of The Quad (Top)
      gl.glVertex3f( 1.0f, 1.0f, 1.0f ); // Bottom
Right Of The Quad (Top)

      gl.glColor3f( 0f,1f,0f ); //green color
      gl.glVertex3f( 1.0f, -1.0f, 1.0f ); // Top
Right Of The Quad
      gl.glVertex3f( -1.0f, -1.0f, 1.0f ); // Top
Left Of The Quad
```

```
      gl.glVertex3f( -1.0f, -1.0f, -1.0f ); //
Bottom Left Of The Quad
      gl.glVertex3f( 1.0f, -1.0f, -1.0f ); //
Bottom Right Of The Quad

      gl.glColor3f( 0f,0f,1f ); //blue color
      gl.glVertex3f( 1.0f, 1.0f, 1.0f ); // Top
Right Of The Quad (Front)
      gl.glVertex3f( -1.0f, 1.0f, 1.0f ); // Top
Left Of The Quad (Front)
      gl.glVertex3f( -1.0f, -1.0f, 1.0f ); //
Bottom Left Of The Quad
      gl.glVertex3f( 1.0f, -1.0f, 1.0f ); //
Bottom Right Of The Quad

      gl.glColor3f( 1f,1f,0f ); //yellow (red +
green)
      gl.glVertex3f( 1.0f, -1.0f, -1.0f ); //
Bottom Left Of The Quad
      gl.glVertex3f( -1.0f, -1.0f, -1.0f ); //
Bottom Right Of The Quad
      gl.glVertex3f( -1.0f, 1.0f, -1.0f ); // Top
Right Of The Quad (Back)
      gl.glVertex3f( 1.0f, 1.0f, -1.0f ); // Top
Left Of The Quad (Back)

      gl.glColor3f( 1f,0f,1f ); //purple (red +
green)
      gl.glVertex3f( -1.0f, 1.0f, 1.0f ); // Top
Right Of The Quad (Left)
      gl.glVertex3f( -1.0f, 1.0f, -1.0f ); // Top
Left Of The Quad (Left)
      gl.glVertex3f( -1.0f, -1.0f, -1.0f ); //
Bottom Left Of The Quad
      gl.glVertex3f( -1.0f, -1.0f, 1.0f ); //
Bottom Right Of The Quad

      gl.glColor3f( 0f,1f, 1f ); //sky blue (blue
+green)
```

```
        gl.glVertex3f( 1.0f, 1.0f, -1.0f ); // Top
Right Of The Quad (Right)
        gl.glVertex3f( 1.0f, 1.0f, 1.0f ); // Top
Left Of The Quad
        gl.glVertex3f( 1.0f, -1.0f, 1.0f ); //
Bottom Left Of The Quad
        gl.glVertex3f( 1.0f, -1.0f, -1.0f ); //
Bottom Right Of The Quad
        gl.glEnd(); // Done Drawing The Quad
        gl.glFlush();

        rquad -= 0.15f;
    }
```

Cube avec des facettes texturées :

```
    public void init(GLAutoDrawable drawable) {

        final GL2 gl = drawable.getGL().getGL2();

        gl.glShadeModel(GL2.GL_SMOOTH);
        gl.glClearColor(0f, 0f, 0f, 0f);
        gl.glClearDepth(1.0f);
        gl.glEnable(GL2.GL_DEPTH_TEST);
        gl.glDepthFunc(GL2.GL_LEQUAL);

gl.glHint(GL2.GL_PERSPECTIVE_CORRECTION_HINT,
GL2.GL_NICEST);

        //
        gl.glEnable(GL2.GL_TEXTURE_2D);
        try{

            File im = new File("E:\\office\\boy.jpg
");
            Texture t = TextureIO.newTexture(im,
true);
            texture= t.getTextureObject(gl);

        }catch(IOException e){
            e.printStackTrace();
        }
```

```
    }
```

```
    public void display(GLAutoDrawable drawable) {

        // TODO Auto-generated method stub
        final GL2 gl = drawable.getGL().getGL2();
        gl.glClear(GL2.GL_COLOR_BUFFER_BIT |
GL2.GL_DEPTH_BUFFER_BIT);
        gl.glLoadIdentity(); // Reset The View
        gl.glTranslatef(0f, 0f, -5.0f);

        gl.glRotatef(xrot, 1.0f, 1.0f, 1.0f);
        gl.glRotatef(yrot, 0.0f, 1.0f, 0.0f);
        gl.glRotatef(zrot, 0.0f, 0.0f, 1.0f);

        gl.glBindTexture(GL2.GL_TEXTURE_2D,
texture);
        gl.glBegin(GL2.GL_QUADS);

        // Front Face
        gl.glTexCoord2f(0.0f, 0.0f); gl.glVertex3f(-
1.0f, -1.0f, 1.0f);
        gl.glTexCoord2f(1.0f, 0.0f); gl.glVertex3f(
1.0f, -1.0f, 1.0f);
        gl.glTexCoord2f(1.0f, 1.0f); gl.glVertex3f(
1.0f, 1.0f, 1.0f);
        gl.glTexCoord2f(0.0f, 1.0f); gl.glVertex3f(-
1.0f, 1.0f, 1.0f);

        // Back Face
        gl.glTexCoord2f(1.0f, 0.0f); gl.glVertex3f(-
1.0f, -1.0f, -1.0f);
        gl.glTexCoord2f(1.0f, 1.0f); gl.glVertex3f(-
1.0f, 1.0f, -1.0f);
        gl.glTexCoord2f(0.0f, 1.0f); gl.glVertex3f(
1.0f, 1.0f, -1.0f);
        gl.glTexCoord2f(0.0f, 0.0f); gl.glVertex3f(
1.0f, -1.0f, -1.0f);

        // Top Face
```

```
      gl.glTexCoord2f(0.0f, 1.0f); gl.glVertex3f(-
1.0f, 1.0f, -1.0f);
      gl.glTexCoord2f(0.0f, 0.0f); gl.glVertex3f(-
1.0f, 1.0f, 1.0f);
      gl.glTexCoord2f(1.0f, 0.0f); gl.glVertex3f(
1.0f, 1.0f, 1.0f);
      gl.glTexCoord2f(1.0f, 1.0f); gl.glVertex3f(
1.0f, 1.0f, -1.0f);

      // Bottom Face
      gl.glTexCoord2f(1.0f, 1.0f); gl.glVertex3f(-
1.0f, -1.0f, -1.0f);
      gl.glTexCoord2f(0.0f, 1.0f); gl.glVertex3f(
1.0f, -1.0f, -1.0f);
      gl.glTexCoord2f(0.0f, 0.0f); gl.glVertex3f(
1.0f, -1.0f, 1.0f);
      gl.glTexCoord2f(1.0f, 0.0f); gl.glVertex3f(-
1.0f, -1.0f, 1.0f);

      // Right face
      gl.glTexCoord2f(1.0f, 0.0f); gl.glVertex3f(
1.0f, -1.0f, -1.0f);
      gl.glTexCoord2f(1.0f, 1.0f); gl.glVertex3f(
1.0f, 1.0f, -1.0f);
      gl.glTexCoord2f(0.0f, 1.0f); gl.glVertex3f(
1.0f, 1.0f, 1.0f);
      gl.glTexCoord2f(0.0f, 0.0f); gl.glVertex3f(
1.0f, -1.0f, 1.0f);

      // Left Face
      gl.glTexCoord2f(0.0f, 0.0f); gl.glVertex3f(-
1.0f, -1.0f, -1.0f);
      gl.glTexCoord2f(1.0f, 0.0f); gl.glVertex3f(-
1.0f, -1.0f, 1.0f);
      gl.glTexCoord2f(1.0f, 1.0f); gl.glVertex3f(-
1.0f, 1.0f, 1.0f);
      gl.glTexCoord2f(0.0f, 1.0f); gl.glVertex3f(-
1.0f, 1.0f, -1.0f);
      gl.glEnd();
      gl.glFlush();
```

```

      //change the speeds here
      xrot += .1f;
      yrot += .1f;
      zrot += .1f;
   }
```

Partie 5 :

Polygone avec éclairage :

```
   public void display( GLAutoDrawable drawable )
{

      final GL2 gl = drawable.getGL().getGL2();
      gl.glColor3f(1f,0f,0f); //applying red

      // Clear The Screen And The Depth Buffer
      gl.glClear( GL2.GL_COLOR_BUFFER_BIT |
      GL2.GL_DEPTH_BUFFER_BIT );
      gl.glLoadIdentity();       // Reset The View
      gl.glRotatef( rpoly, 0.0f, 1.0f, 0.0f );

      gl.glBegin( GL2.GL_POLYGON );

      gl.glVertex3f( 0f,0.5f,0f );
      gl.glVertex3f( -0.5f,0.2f,0f );
      gl.glVertex3f( -0.5f,-0.2f,0f );
      gl.glVertex3f( 0f,-0.5f,0f );
      gl.glVertex3f( 0f,0.5f,0f );
      gl.glVertex3f( 0.5f,0.2f,0f );
      gl.glVertex3f( 0.5f,-0.2f,0f );
      gl.glVertex3f( 0f,-0.5f,0f );

      gl.glEnd();

      gl.glFlush();

      rpoly += 0.2f;  //assigning the angle
      gl.glEnable( GL2.GL_LIGHTING );
      gl.glEnable( GL2.GL_LIGHT0 );
      gl.glEnable( GL2.GL_NORMALIZE );

      // weak RED ambient
      float[] ambientLight = { 0.1f, 0.f, 0.f,0f
};
      gl.glLightfv(GL2.GL_LIGHT0, GL2.GL_AMBIENT,
ambient-Light, 0);

      // multicolor diffuse
      float[] diffuseLight = { 1f,2f,1f,0f };
      gl.glLightfv( GL2.GL_LIGHT0, GL2.GL_DIFFUSE,
diffuse-Light, 0 );
   }
```

Les solutions complètes peuvent être trouvées sur les liens ci-dessous :

1- Configuration OpenGL avec NetBeans :
https://www.youtube.com/watch?v=t0ntwdDqqH4

2- Dessin des lignes + Formes prédéfinis :
https://www.tutorialspoint.com/jogl/drawing_with_gl_lines.htm

https://www.tutorialspoint.com/jogl/jogl_3d_basics.htm

https://www.tutorialspoint.com/jogl/jogl_pre_defined_shapes.htm

3- Dessin des triangles + Coloration :
https://www.tutorialspoint.com/jogl/jogl_coloring.htm

https://www.tutorialspoint.com/jogl/jogl_3d_triangle.htm

4- Transformations / Translation :
https://www.tutorialspoint.com/jogl/jogl_transformation.htm

5- Transformations / Changement d’échelle :
https://www.tutorialspoint.com/jogl/jogl_scaling.htm

6- Transformations / Rotation :
https://www.tutorialspoint.com/jogl/jogl_rotation.htm

7- Dessiner un cube + Texture :
https://www.tutorialspoint.com/jogl/jogl_3d_cube.htm

8- Eclairage :

https://www.tutorialspoint.com/jogl/jogl_lighting.htm