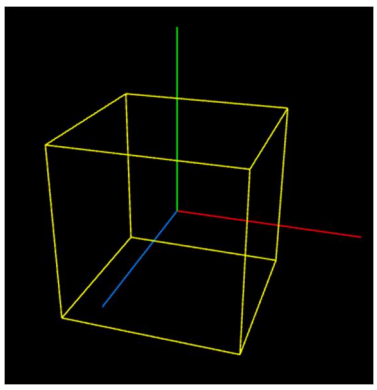
＜本日の課題＞

課題 4-1　3 次元図形の描画（１）

次に示す WebGL のサンプルコード **sample04-1** を実行し，図に示すような座標系の原点を中心とした立方体が表示されることを確認しなさい。また，透視投影の設定やカメラの視点設定の値を変えることでどのような影響があるかを確認しなさい。

sample04-1(抜粋)

```
gl.matrixMode( gl.PROJECTION );    // 投影行列の操作モード
gl.loadIdentity();                 // 行列を単位行列に初期化
// カメラの視界設定（透視投影）
gl.perspective(45.0,               // 視野角
               window.innerWidth / window.innerHeight,   // アスペクト比(w/h)
               0.1, 100.0);        // カメラとの最短距離，最長距離

gl.matrixMode( gl.MODELVIEW );     // モデルビュー行列の操作モード
gl.loadIdentity();                 // 行列を単位行列に初期化
// カメラの視点設定（位置，向き）
gl.lookAt( 1.0, 1.5, 3.0,          // カメラ位置
           0.0, 0.0, 0.0,          // 注視点
           0.0, 1.0, 0.0 );        // 上方向ベクトル（y 軸）

  ～ 中略 ～

//立方体の描画
gl.pushMatrix();
  gl.color(1.0, 1.0, 0.0, 1.0);
  // 前面
  gl.begin(gl.LINE_LOOP);
    gl.vertex(-0.5, -0.5,  0.5);
    gl.vertex( 0.5, -0.5,  0.5);
    gl.vertex( 0.5,  0.5,  0.5);
    gl.vertex(-0.5,  0.5,  0.5);
  gl.end();
  // 背面
  gl.begin(gl.LINE_LOOP);
    gl.vertex(-0.5, -0.5, -0.5);
    gl.vertex( 0.5, -0.5, -0.5);
    gl.vertex( 0.5,  0.5, -0.5);
    gl.vertex(-0.5,  0.5, -0.5);
  gl.end();
  // 側面
  gl.begin(gl.LINES);
    gl.vertex(-0.5, -0.5,  0.5);
    gl.vertex(-0.5, -0.5, -0.5);

    gl.vertex( 0.5, -0.5,  0.5);
    gl.vertex( 0.5, -0.5, -0.5);

    gl.vertex( 0.5,  0.5,  0.5);
    gl.vertex( 0.5,  0.5, -0.5);

    gl.vertex(-0.5,  0.5,  0.5);
    gl.vertex(-0.5,  0.5, -0.5);
  gl.end();
gl.popMatrix();
```



補足）window.innerWidth ：表示領域の幅，window.innerHeight：表示領域の高さ（JavaScript プロパティ）

課題 4-2　関数の導入

立方体の描画部分を関数化したサンプルコード **sample04-2** を実行し，サンプルコード **sample04-1** と同様の出力結果となることを確認しなさい。また，座標軸の描画部分を同様に関数化し，その動作を確認しなさい。

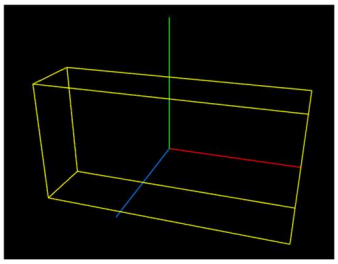
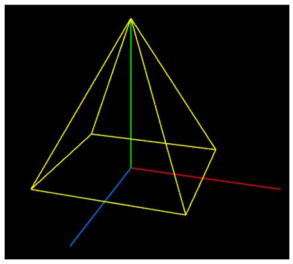
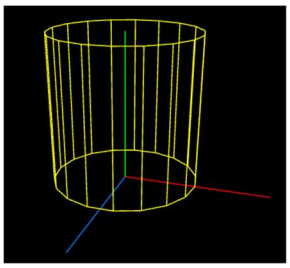
課題 4-3　立体の描画（１）

次に示す立体を描画しなさい。

1）幅 2.0，高さ 1.0，奥行き 0.5 の直方体（Box）

2）高さ 1.0，底面が 1.0×1.0 の四角錐（Pyramid）

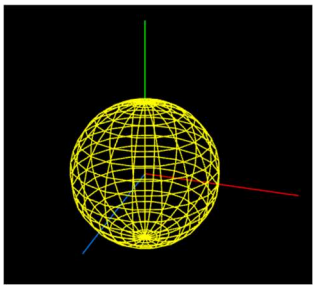
3）高さ 1.0，半径 0.5 の円柱（Cylinder）　（側面の辺の数は，円柱と分かる範囲で任意とする）

Hint：円柱は，xz平面(y=0)とy=1 の面に円を描き，円周上に等間隔にとった点同士を縦線で結ぶ。

○課題 4-4　立体の描画（２）

半径 0.5 の球（Sphere）を描画しなさい。なお，格子間隔は，球と分かる範囲で任意とする。



Hint: 球は，経線（縦線）と緯線（横線）を描画することにより表現する。

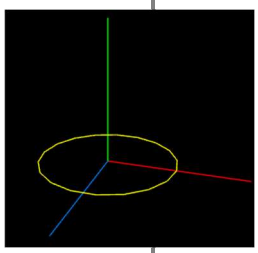
・経線：yz平面に円を描き，それをy軸周りに少しずつ回転・描画させながら 180°まで回転させる

・緯線：球を横方向にスライスしたときにできる輪の形（円）を，高さを変えながら描画する。

参考）円の描画：xz平面に円を描くサンプルコード，円周上に等間隔に点をとりそれらを線で結ぶ

sample04-3(関数抜粋)

```
function drawCircle( r ) {     //引数 r は円の半径
    const segments = 18;       //円周の分割数（大きいほど滑らかな円）
    gl.begin(gl.LINE_LOOP);
    for (let i = 0; i <= segments; i++) {
        let theta = (i / segments) * 2 * Math.PI;  // 角度を計算
        let x = r * Math.cos(theta);          // x 座標
        let z = r * Math.sin(theta);          // z 座標
        gl.vertex(x, 0.0, z);                 // xz 平面なので y=0
    }
    gl.end();
}
```



課題 4-5　立体の座標変換

課題 4-3, 4-4 で作成した立体から一つ選び，次の図形変換を順番に与え，変換後の図形を表示しなさい。

1）立体をx軸に沿って正方向に 0.5，z軸に沿って正方向に 1.0 だけ平行移動しなさい。
2）立体をy軸周りに 45° 回転しなさい。
3）立体をy軸方向に 2 倍拡大しなさい。

課題 4-6　カメラの視点変更

課題 4-3, 4-4 で作成した立体を選び，以下の条件に従いカメラを設定しなさい。その際の視点の変更による立体の見え方の違いを確認しなさい。

・視野角を 60 度に設定
・カメラの位置を（3.0, 3.0, 3.0），注視点を原点（0.0, 0.0, 0.0）に設定
・上方向ベクトルをz軸方向に設定

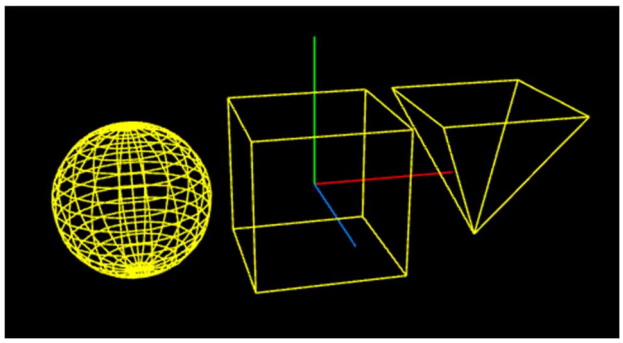
課題 4-7　投影方法の比較

以下の手順でそれぞれ図形を描画し, 透視投影と平行投影の違いを確認しなさい。なお, 各投影の設定値は, 図形が見える範囲で任意とする。

1）`gl.perspective()`を使用し，透視投影で立方体を描画する。
~~2）`gl.ortho()`を使用し，平行投影で同じ立方体を描画する。~~

〇課題 4-8　図形の配置

次に示す図に示すようなシーンを描画しなさい。ここで, 立体の大きさや相対的な寸法, 間隔は任意とする。



以上