

Exercice 1

Une pression sur la touche 'l' met le polygone en mode ligne.

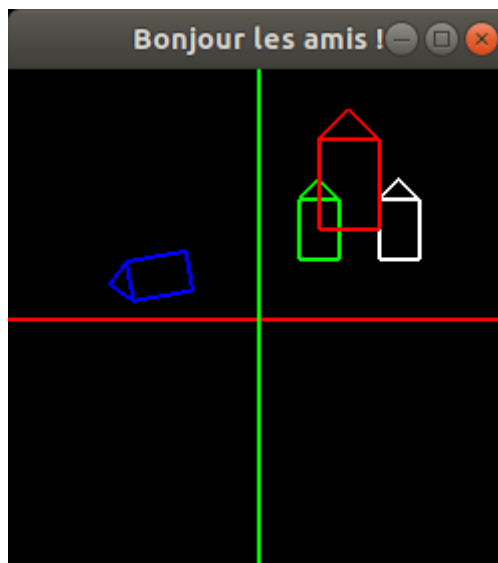
Une pression sur la touche 'r' met le polygone en mode remplie.

Une pression sur la touche 'p' met le polygone en mode points.

Exercice 2

Les transformations sont appliquées les unes à la suite des autres. Si l'on applique plusieurs transformations à des objets différents toutes les transformations faites précédemment sont également exécuté.

LoadIdentity sert à repartir de l'axe d'origine. Les transformations ne sont donc pas enchainées à la suite.



Scale augmente la taille de l'image et Rotate la tourne. Quant à translate il la déplace.

Exercice 3

Code :

```
double * translation(double x, double y, double z){
    double * T = new double[16];
    T[0] = 1;   T[1] = 0;   T[2] = 0;   T[3] = 0;
    T[4] = 0;   T[5] = 1;   T[6] = 0;   T[7] = 0;
    T[8] = 0;   T[9] = 0;   T[10] = 1;  T[11] = 0;
    T[12] = x;  T[13] = y;  T[14] = z;  T[15] = 1;
    return T;
}
```

```

double * scale(double x, double y, double z){
    double * S = new double[16];
    S[0] = x;    S[1] = 0;    S[2] = 0;    S[3] = 0;
    S[4] = 0;    S[5] = y;    S[6] = 0;    S[7] = 0;
    S[8] = 0;    S[9] = 0;    S[10] = z;   S[11] = 0;
    S[12] = 0;   S[13] = 0;   S[14] = 0;   S[15] = 1;
    return S;
}

double * rotation(double angle, double x, double y, double z){
    double * R = new double[16];
    R[0] = cos(angle*PI/180);    R[1] = sin(angle*PI/180);    R[2] = 0;    R[3] = 0;
    R[4] = -sin(angle*PI/180);    R[5] = cos(angle*PI/180);    R[6] = 0;    R[7] = 0;
    R[8] = 0;    R[9] = 0;    R[10] = 1;    R[11] = 0;
    R[12] = 0;    R[13] = 0;    R[14] = 0;    R[15] = 1;
    return R;
}

```

Main :

```

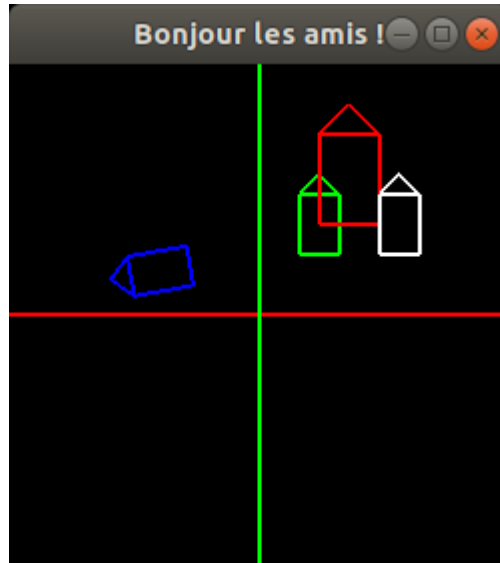
glLoadIdentity();
glColor3f(0.0,0.0,1.0);
glLoadMatrixd(rotation(100,0,0,1));
draw();

glLoadIdentity();
glLoadMatrixd(scale(1.5,1.5,1));
glColor3f(1.0,0.0,0.0);
draw();

glLoadIdentity();
glLoadMatrixd(translation(4, 0, 0));
glColor3f(2.0,2.0,1.0);
draw();

```

On obtient le résultat suivant avec les méthodes créées. Qui est une copie parfaite de ce que faisait l'exercice 2 avec les mêmes valeurs.



Code du compound :

```
double * compose(double * M1, double * M2){
    double * C = new double[16];
    C[0] = M1[0]*M2[0] + M1[4]* M2[1] + M1[8]* M2[2] + M1[12] * M2[3];
    C[1] = M1[1]*M2[0] + M1[5]* M2[1] + M1[9]* M2[2] + M1[13] * M2[3];
    C[2] = M1[2]*M2[0] + M1[6]* M2[1] + M1[10]* M2[2] + M1[14] * M2[3];
    C[3] = M1[3]*M2[0] + M1[7]* M2[1] + M1[11]* M2[2] + M1[15] * M2[3];

    C[4] = M1[0]*M2[4] + M1[4]* M2[5] + M1[8]* M2[6] + M1[12] * M2[7];
    C[5] = M1[1]*M2[4] + M1[5]* M2[5] + M1[9]* M2[6] + M1[13] * M2[7];
    C[6] = M1[2]*M2[4] + M1[6]* M2[5] + M1[10]* M2[6] + M1[14] * M2[7];
    C[7] = M1[3]*M2[4] + M1[7]* M2[5] + M1[11]* M2[6] + M1[15] * M2[7];

    C[8] = M1[0]*M2[8] + M1[4]* M2[9] + M1[8]* M2[10] + M1[12] * M2[11];
    C[9] = M1[1]*M2[8] + M1[5]* M2[9] + M1[9]* M2[10] + M1[13] * M2[11];
    C[10] = M1[2]*M2[8] + M1[6]* M2[9] + M1[10]* M2[10] + M1[14] * M2[11];
    C[11] = M1[3]*M2[8] + M1[7]* M2[9] + M1[11]* M2[10] + M1[15] * M2[11];

    C[12] = M1[0]*M2[12] + M1[4]* M2[13] + M1[8]* M2[14] + M1[12] * M2[15];
    C[13] = M1[1]*M2[12] + M1[5]* M2[13] + M1[9]* M2[14] + M1[13] * M2[15];
    C[14] = M1[2]*M2[12] + M1[6]* M2[13] + M1[10]* M2[14] + M1[14] * M2[15];
    C[15] = M1[3]*M2[12] + M1[7]* M2[13] + M1[11]* M2[14] + M1[15] * M2[15];

    return C;
}
```