```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class MyTransform
6  {
7
8      public static Matrix4By4 YawRotate(float RotX)
9      {
10
11         Matrix4By4 YawMatrix = new Matrix4By4(new MyVector3(Mathf.Cos(RotX),
             0, -Mathf.Sin(RotX)),
12                                               new MyVector3(0, 1, 0),
13                                               new MyVector3(Mathf.Sin(RotX),
                       0, Mathf.Cos(RotX)),
14                                               MyVector3.Zero());
15
16         return YawMatrix;
17     }
18     public static Matrix4By4 PitchRotate(float RotY)
19     {
20         Matrix4By4 PitchMatrix = new Matrix4By4(new MyVector3(1, 0, 0),
21                                                 new MyVector3(0, Mathf.Cos
                       (RotY), Mathf.Sin(RotY)),
22                                                 new MyVector3(0, -Mathf.Sin
                       (RotY), Mathf.Cos(RotY)),
23                                                 MyVector3.Zero());
24
25         return PitchMatrix;
26     }
27     public static Matrix4By4 RollRotate(float RotZ)
28     {
29         Matrix4By4 RollMatrix = new Matrix4By4(new MyVector3(Mathf.Cos(RotZ),
              Mathf.Sin(RotZ), 0),
30                                                new MyVector3(-Mathf.Sin(RotZ),
                       Mathf.Cos(RotZ), 0),
31                                                new MyVector3(0, 0, 1),
32                                                MyVector3.Zero());
33         return RollMatrix;
34     }
35     public static Matrix4By4 Rotation(float RotX, float RotY, float RotZ)
36     {
37         Matrix4By4 RotationMatrix = YawRotate(RotX) * (PitchRotate(RotY) *
             RollRotate(RotZ));
38
39
40         return RotationMatrix;
41     }
42     public static Matrix4By4 Scale(float ScaleX, float ScaleY, float ScaleZ)
43     {
44         Matrix4By4 ScaleMatrix = new Matrix4By4(new MyVector3(ScaleX, 0, 0) ,
              new MyVector3(0, ScaleY, 0) , new MyVector3(0, 0, ScaleZ),
             MyVector3.Zero());
```

```
45
46
47          return ScaleMatrix;
48      }
49      public static Matrix4By4 Translate(float TranslateX, float TranslateY,
          float TranslateZ)
50      {
51          Matrix4By4 TranslateMatrix = new Matrix4By4(new MyVector3(1, 0, 0),
              new MyVector3(0,1, 0), new MyVector3(0, 0, 1), new MyVector3
              (TranslateX, TranslateY, TranslateZ));
52          return TranslateMatrix;
53      }
54      public static Matrix4By4 TRS(Matrix4By4 Translate, Matrix4By4 Rotate,
          Matrix4By4 Scale)
55      {
56
57          Matrix4By4 M =  (Rotate * Scale)* Translate;
58
59          return M;
60      }
61      public static Matrix4By4 QuaternionToRotationMatrix(MyVector3 quat)
62      {
63          Matrix4By4 Rotation = MyTransform.Rotation(quat.x, quat.y, quat.z);
64
65
66          return Rotation;
67      }
68      //public static Matrix4By4 ScaleInverse(float ScaleX, float ScaleY, float
           ScaleZ)
69      //{
70
71      //}
72      //public static Matrix4By4 RotationInverse()
73      //{
74      //    return new Matrix4By4(GetRow(),GetRow(1), GetRow(1), GetRow(1))
75      //}
76      //public static Matrix4By4 TranslateInverse(float TranslateX, float
          TranslateY, float TranslateZ)
77      //{
78
79      //}
80  }
```