```
 1  using System.Collections;
 2  using System.Collections.Generic;
 3  using NUnit.Framework;
 4  using UnityEngine;
 5  using System;
 6  using UnityEngine.TestTools;
 7
 8  public class MatrixTests
 9  {
10      [Test]
11      public void MatrixMultiply1()
12      {
13          double[,] aValues = new double[2, 2] { { 4, -1 }, { 0, 7 } };
14          Matrix a = new Matrix(aValues);
15          double[,] bValues = new double[2, 2] { { -1, 1 }, { 5, 2 } };
16          Matrix b = new Matrix(bValues);
17          double[,] cValues = new double[2, 2] { { -9, 2 }, { 35, 14 } };
18          Matrix c = new Matrix(cValues);
19          Assert.IsTrue(Matrix.Equal(Matrix.Multiply(a, b), c));
20      }
21
22      [Test]
23      public void MatrixMultiply2()
24      {
25          double[,] aValues = new double[2, 2] { { 2.1, 3.5 }, { -0.5, 6 } };
26          Matrix a = new Matrix(aValues);
27          double[,] bValues = new double[2, 2] { { 0.5, 3 }, { -6, 6 } };
28          Matrix b = new Matrix(bValues);
29          double[,] cValues = new double[2, 2] { { -19.95, 27.3 }, { -36.25,
              34.5 } };
30          Matrix c = new Matrix(cValues);
31          Assert.IsTrue(Matrix.Equal(Matrix.Multiply(a, b), c));
32      }
33
34      [Test]
35      public void MatrixMultiply3()
36      {
37          double[,] aValues = new double[2, 3] { { 2, 4, -1 }, { 2, -6, 5 } };
38          Matrix a = new Matrix(aValues);
39          double[,] bValues = new double[3, 2] { { 3, 0 }, { 7, -1 }, { -9,
              11} };
40          Matrix b = new Matrix(bValues);
41          double[,] cValues = new double[2, 2] { { 43, -15 }, { -81, 61 } };
42          Matrix c = new Matrix(cValues);
43          Assert.IsTrue(Matrix.Equal(Matrix.Multiply(a, b), c));
44      }
45
46      [Test]
47      public void MatrixMultiply4()
48      {
49          double[,] aValues = new double[3, 2] { { 2, 3}, { -2, 4 }, { 1,
              8 } };
50          Matrix a = new Matrix(aValues);
```

```
51          double[,] bValues = new double[3, 3] { { -4, 2, -1 }, { -5, -1, 0 },
               { 2, -6, 3} };
52          Matrix b = new Matrix(bValues);
53          Assert.That(() => Matrix.Multiply(a, b), Throws.TypeOf<InvalidSize>
              ());
54      }
55
56      [Test]
57      public void MatrixAdd1()
58      {
59          double[,] aValues = new double[2, 2] { { 4, -1 }, { 0, 7 } };
60          Matrix a = new Matrix(aValues);
61          double[,] bValues = new double[2, 2] { { -1, 1 }, { 5, 2 } };
62          Matrix b = new Matrix(bValues);
63          double[,] cValues = new double[2, 2] { { 3, 0 }, { 5, 9 } };
64          Matrix c = new Matrix(cValues);
65          Assert.IsTrue(Matrix.Equal(Matrix.Add(a, b), c));
66      }
67
68      [Test]
69      public void MatrixAdd2()
70      {
71          double[,] aValues = new double[2, 2] { { 2.1, 3.5 }, { -0.5, 6 } };
72          Matrix a = new Matrix(aValues);
73          double[,] bValues = new double[2, 2] { { 0.5, 3 }, { -6, 6 } };
74          Matrix b = new Matrix(bValues);
75          double[,] cValues = new double[2, 2] { { 2.6, 6.5 }, { -6.5, 12 } };
76          Matrix c = new Matrix(cValues);
77          Assert.IsTrue(Matrix.Equal(Matrix.Add(a, b), c));
78      }
79
80      [Test]
81      public void MatrixAdd3()
82      {
83          double[,] aValues = new double[2, 3] { { 2, 4, -1 }, { 2, -6, 5 } };
84          Matrix a = new Matrix(aValues);
85          double[,] bValues = new double[3, 2] { { 3, 0 }, { 7, -1 }, { -9,
              11 } };
86          Matrix b = new Matrix(bValues);
87          Assert.That(() => Matrix.Add(a, b), Throws.TypeOf<InvalidSize>());
88      }
89
90      [Test]
91      public void EmptyMatrix()
92      {
93          Matrix a = new Matrix(3, 2);
94          double[,] bValues = new double[3, 2] { { 0, 0 }, { 0, 0 }, { 0,
              0 } };
95          Matrix b = new Matrix(bValues);
96          Assert.IsTrue(Matrix.Equal(a, b));
97      }
98
99      [Test]
```

```
100     public void GetColumn1()
101     {
102         double[,] aValues = new double[2, 3] { { 2, 4, -1 }, { 2, -6, 5 } };
103         Matrix a = new Matrix(aValues);
104         double[] bValues = { 4, -6 };
105         Assert.IsTrue(a.GetColumn(1)[0] == bValues[0] && a.GetColumn(1)[1]
              == bValues[1]);
106     }
107
108     [Test]
109     public void GetColumn2()
110     {
111         double[,] aValues = new double[2, 3] { { 2, 4, -1 }, { 2, -6, 5 } };
112         Matrix a = new Matrix(aValues);
113         double[] bValues = { 2, 2 };
114         Assert.IsTrue(a.GetColumn(0)[0] == bValues[0] && a.GetColumn(0)[1]
              == bValues[1]);
115     }
116
117     [Test]
118     public void GetColumn3()
119     {
120         double[,] aValues = new double[2, 3] { { 2, 4, -1 }, { 2, -6, 5 } };
121         Matrix a = new Matrix(aValues);
122         double[] bValues = { -1, 5 };
123         Assert.IsTrue(a.GetColumn(2)[0] == bValues[0] && a.GetColumn(2)[1]
              == bValues[1]);
124     }
125
126     [Test]
127     public void GetColumn4()
128     {
129         double[,] aValues = new double[2, 3] { { 2, 4, -1 }, { 2, -6, 5 } };
130         Matrix a = new Matrix(aValues);
131         Assert.That(() => a.GetColumn(3), Throws.TypeOf<OutOfBounds>());
132     }
133 }
```