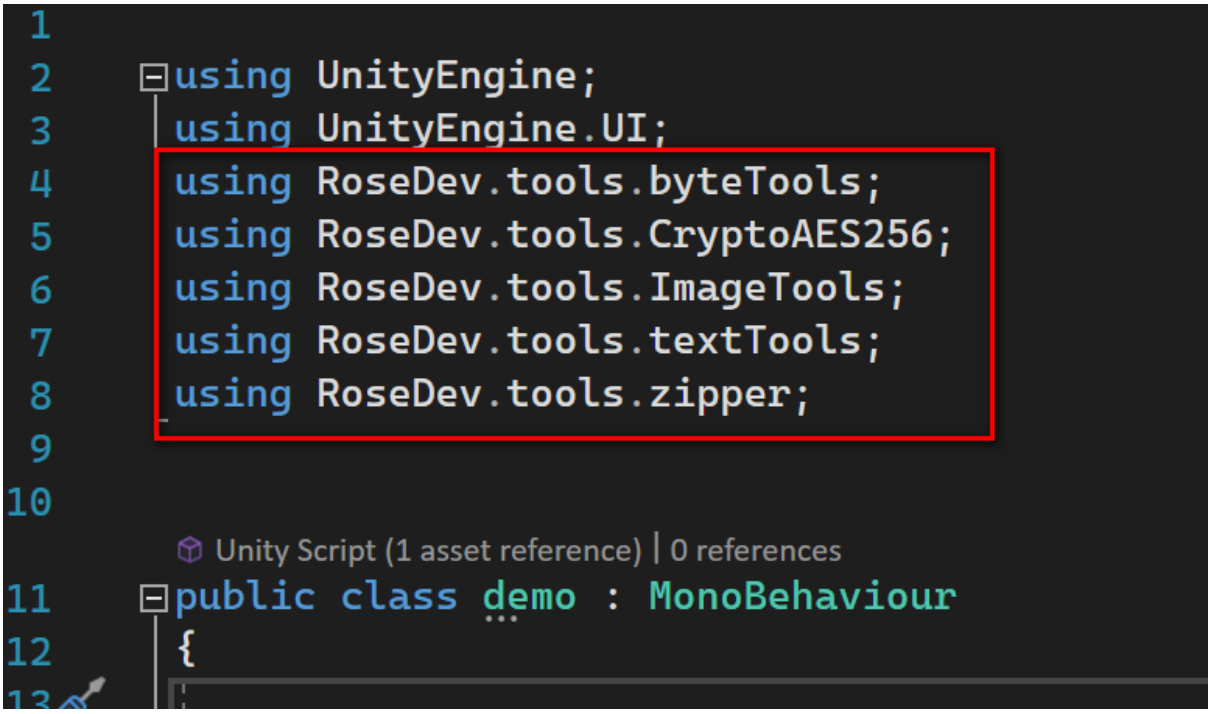
list :

- To use the following scripts, we must first add the required libraries to our script as shown below:



byte tool

- If you have a long byte array and want to send it to the client, the best way is to divide it into several smaller byte arrays. In the following code, we divide the bigByteArray into several parts by calling the BufferSplit method and store it in the buf[][] array.

```
#region byte tools
[SerializeField] byte[] bigByteArray;
[SerializeField] byte[][] buf;
// for split big array you can use this method
0 references
public void byteToolExample_splitByte()
{
    buf = byteTools.BufferSplit(bigByteArray, 512);
    Debug.Log(buf.Length);
}
0 references
public void byteToolExample_MergeTowToOne()
{
    bigByteArray = byteTools.MergeTowToOne(buf);
    Debug.Log(bigByteArray.Length);
}

#endregion
```

size of parts

- We also use the following code to merge split presentations.

```
public void byteToolExample_MergeTowToOne()
{
    bigByteArray = byteTools.MergeTowToOne(buf);
    Debug.Log(bigByteArray.Length);
}
```

Crypto (encryption and decryption)

- If you have a string and want to encrypt or decrypt it, you can call the following method (note that your password must be the same when encrypting or decrypting).

```
#region crypto
// for encrypt and decrypt any string use this methods
0 references
public void cryptoExmaple_Encrypt(string str)
{
    string EncryptStr = AEScrypto.EncryptText(str, "123abc");
    Debug.Log(EncryptStr);
}
0 references
public void cryptoExmaple_Decrypt(string str)
{
    string DecryptStr = AEScrypto.DecryptText(str, "123abc");
    Debug.Log(DecryptStr);
}
#endregion
```

image tool

- To serialize or deserialize an image, it is enough to write the path of the desired file in the following method. The output of this method will be a string containing the serialized image information.

```
#region Image Serialize
string SerializedImage;
[SerializeField] Image img;
0 references
public void ImageToolsExample_Serialize()
{
    string url_ = "file path";
    Vector2 imageSize = new Vector2(256, 256);
    SerializedImage = imageConverter.Serialize(url_, imageSize);
}
0 references
public void ImageToolsExample_Deserialize()
{
    Vector2 imageSize = new Vector2(256, 256);
    img.sprite = imageConverter.DeSerialize(SerializedImage, imageSize);
}
#endregion
```

text Tool

- If you need to display a large number, you can use "K" for a thousand and "M" for a million, how to use it is as follows.

```
#region text tool
0 references
public void convertBigNumbersTo_K_M()
{
    string str = textTool.numberFormat(10000000);
    Debug.Log(str); // print : 1M
}
#endregion
```

zipper (String compressor)

- Suppose we serialize an image and the result is a very long string, now if we convert this long string to a byte in the usual way, the result will be a very long array even if our image size is small, so we We can use the zipper method and convert a long string to a compressed byte array.

```
0 references
public void CompressStringToByte_DecompressByteToString()
{
    string str = "test mess";

    byte[] b = zipper.CompressStringToByte(str);
    Debug.Log( b.Length);

    str = zipper.DecompressByteToString (b);
    Debug.Log(str);
}
```