Code Examples

Key Points

- **Interfaces enforce functionality** when they are implemented, meaning that a class must implement versions of the methods and members declared in the interface.
- This allows generalisation, where different classes can be treated as if they are the same, even though they're not.
- A common use would be an **IUsable** interface, which allows different 'usable' object to be used with one trigger.
- They can also be used as communication interfaces between scripts to enforce **abstraction**.

Code Examples

IUsable Interface

```
1. using UnityEngine;
2.
3. public interface IUsable
4. {
5.     public void Use();
6. }
```

Lever implementing the IUsable interface

```
1. using UnityEngine;
2.
3. public class Lever : MonoBehaviour, IUsable
4. {
5.     public void Use()
```

```
6.     {
7.         Debug.Log("The Lever was used!");
8.     }
9.  }
```