Code Examples

Composite Direction (using Clamp Magnitude)

```
1. private void Update()
2. {
3.     float x = Input.GetAxis("Horizontal");
4.     float z = Input.GetAxis("Vertical");
5.
6.     Vector3 movement = new Vector3(x, 0, z);
7.     movement = Vector3.ClampMagnitude(movement, 1);
8.     transform.Translate(movement * speed * Time.deltaTime);
9. }
```

Basic Lerp Example

```
1. using UnityEngine;
2. using System.Collections;
3.
4. public class LerpExample : MonoBehaviour
5. {
6.     public float valueToFade = 0;
7.
8.     private void Start()
9.     {
10.        // Fade the value to one over 5 seconds
11.        StartCoroutine(FadeValue(1, 5));
12.    }
13.
14.    IEnumerator FadeValue(float targetValue, float duration)
```

```
    {
        float time = 0;
        float start = valueToFade;

        while (time < duration)
        {
            valueToFade = Mathf.Lerp(start, targetValue, time / duration);
            time += Time.deltaTime;
            yield return null;
        }

        valueToFade = targetValue;
    }
}
```

Move Towards

```
public float valueToChange = 0;
public float rateOfChange = 5;

private void Start()
{
    // Fades the value to 10 at 5 units per second.
    StartCoroutine(ChangeValue(10));
}

IEnumerator ChangeValue(float target)
{
```

```
12.    while (valueToChange != target)
13.    {
14.        valueToChange = Mathf.MoveTowards(valueToChange, target, rateOfChange *
       Time.deltaTime);
15.        yield return null;
16.    }
17. }
```

Lerp Easing

```
1.  Vector3 targetPosition;
2.
3.  private void Update()
4.  {
5.      transform.position = Vector3.Lerp(transform.position, targetPosition,
    Time.deltaTime);
6.  }
```

Smooth Damp

```
1.  public Transform targetPosition;
2.  Vector3 currentVelocity;
3.
4.  private void Update()
5.  {
6.      transform.position = Vector3.SmoothDamp(transform.position,
7.      targetPosition.position, ref currentVelocity, 0.5f);
8.  }
```

Follow target value

```
1.  float hp;
2.  float hpDisplay;
```

```
3.  float changeRate = 50;
4.
5.  private void Start()
6.  {
7.      TakeDamage(50);
8.  }
9.
10. void TakeDamage(float damage)
11. {
12.     hp -= damage;
13.     StopAllCoroutines();
14.     StartCoroutine(ChangeHealth());
15. }
16.
17. IEnumerator ChangeHealth()
18. {
19.     while (hpDisplay != hp)
20.     {
21.         hpDisplay = Mathf.MoveTowards(hpDisplay, hp, changeRate * Time.deltaTime);
22.         yield return null;
23.     }
24. }
```