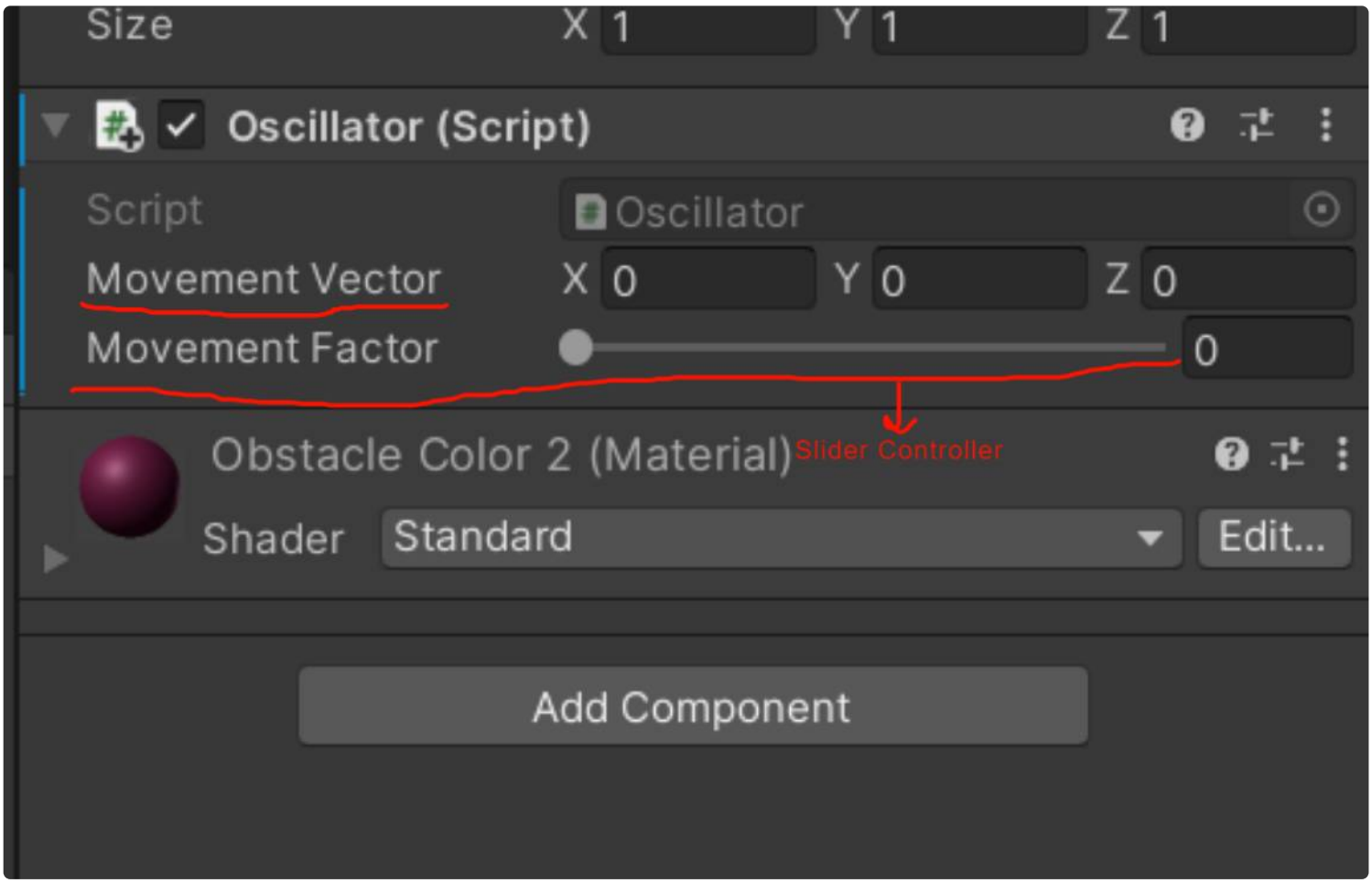
```
0 references
public class Oscillator : MonoBehaviour
{
    2 references
    Vector3 StartingPosition;
    1 reference
    [SerializeField] Vector3 movementVector;
    1 reference
    [SerializeField] [Range(0,1)] float movementFactor;

    0 references
    void Start()
    {
        StartingPosition = transform.position;    // transform.position gives the current position of the game object
    }


    0 references
    void Update()
    {
        Vector3 offset = movementVector * movementFactor;    // direction of movement by a certain factor
        transform.position = StartingPosition + offset;      // new position of the object
    }
}
```

gives the slider controller

Current position of the object



```
float movementFactor;
[SerializeField] float period = 2f;

// Start is called before the first frame update
void Start()
{
    startingPosition = transform.position;
}

// Update is called once per frame
void Update()
{
    float cycles = Time.time / period;  // continually growing over time

    const float tau = Mathf.PI * 2;  // constant value of 6.283
    float rawSinWave = Mathf.Sin(cycles * tau);  // going from -1 to 1

    movementFactor = (rawSinWave + 1f) / 2f;   // recalculated to go from 0 to 1 so its cleaner

    Vector3 offset = movementVector * movementFactor;
    transform.position = startingPosition + offset;
}
```

To prevent from NaN error i.e. attempt to divide by zero in case period is zero.

if(period == 0)

{ return; }

or

if(period <= Mathf.Epsilon) // epsilon is smallest value. you know what epsilon is.

{ return; }