

### **Grid movement with grid size settings. Lclip0001.**

[https://youtu.be/A33\\_quNS0pk](https://youtu.be/A33_quNS0pk)

```
///C# Unity library by René Pol///  
///grid movement with grid size setting (2D)///  
///Tested with unity 2019.3.1.f1///  
  
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
public class Gridmovement : MonoBehaviour  
{  
  
    public float moveSpeed;  
    public float Horizontalgrid;  
    public float Verticalgrid;  
  
    public Transform movePoint;  
    public LayerMask whatStopsMovement;  
  
    void Start()  
    {  
        movePoint.parent = null;  
    }  
  
    void Update()  
    {  
        transform.position = Vector3.MoveTowards(transform.position, movePoint.position, moveSpeed  
        * Time.deltaTime);  
        if (Vector3.Distance(transform.position, movePoint.position) <= .05f)  
        {  
            if (Mathf.Abs(Input.GetAxisRaw("Horizontal")) == 1f)  
            {  
                if (!Physics2D.OverlapCircle(movePoint.position + new  
                Vector3(Input.GetAxisRaw("Horizontal"), 0f), .2f, whatStopsMovement))  
                {  
                    movePoint.position += new Vector3(Horizontalgrid * Input.GetAxisRaw("Horizontal"), 0f);  
                }  
            }  
            else  
            if (Mathf.Abs(Input.GetAxisRaw("Vertical")) == 1f)  
            {  
                if (!Physics2D.OverlapCircle(movePoint.position + new Vector3(0f, 0.9f *  
                Input.GetAxisRaw("Vertical"), 0f), .2f, whatStopsMovement))  
                {  
                    movePoint.position += new Vector3(0f, Verticalgrid * Input.GetAxisRaw("Vertical"), 0f);  
                }  
            }  
        }  
    }  
}
```

*Give the gameobject an empty child object (MovePoint) and drag it into the inspectors window. Set the the distance to go (Grid sizes) also in the inspectors window. You can use a layer that will make the movementstop whenever the Movepoint detect it is within range.*

### **Shrink and grow game objects. Lclip0002.**

<https://youtu.be/OVdwB3ToNwU>

```
///  
C# Unity library by René Pol///  
///  
Shrink and grow game objects (3D-2D)///  
///  
Tested with unity 2019.3.1.f1///  
  
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
public class Growshrink : MonoBehaviour  
{  
  
    public bool growshrink = true;  
  
    public float growRate = 3f;  
    public float shrinkRate = -3f;  
  
    void Update()  
    {  
        if (growshrink == true && transform.localScale.x < 3)  
        {  
            transform.localScale += new Vector3(0.1f, .1f, .1f) * growRate * Time.deltaTime;  
        }  
        if (growshrink == false && transform.localScale.x > 0.5f)  
        {  
            transform.localScale += new Vector3(0.1f, .1f, .1f) * shrinkRate * Time.deltaTime;  
        }  
    }  
}
```

*In this example a public bool is used to choose grow or shrink. You can make separate functions of them that can be called up any time it's needed. You can change the Rate settings in the inspectors window.*

### **Fade Objects in and out. Lclip0003.**

<https://youtu.be/mQdFGXIDzql>

```
///C# Unity library by René Pol///  
///Fade in and out game objects (3D-2D)///  
///Tested with unity 2019.3.1.f1///  
  
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
public class Fading : MonoBehaviour  
{  
    public bool fadeoutin = true;  
    public float fadeSpeed;  
  
    IEnumerator FadeInObject()  
    {  
        while (this.GetComponent<Renderer>().material.color.a < 1)  
        {  
            Color objectColor = this.GetComponent<Renderer>().material.color;  
            float fadeAmount = objectColor.a + (fadeSpeed * Time.deltaTime);  
            objectColor = new Color(objectColor.r, objectColor.g, objectColor.b, fadeAmount);  
            this.GetComponent<Renderer>().material.color = objectColor;  
            yield return null;  
        }  
    }  
  
    IEnumerator FadeOutObject()  
    {  
        while (this.GetComponent<Renderer>().material.color.a > 0)  
        {  
            Color objectColor = this.GetComponent<Renderer>().material.color;  
            float fadeAmount = objectColor.a - (fadeSpeed * Time.deltaTime);  
            objectColor = new Color(objectColor.r, objectColor.g, objectColor.b, fadeAmount);  
            this.GetComponent<Renderer>().material.color = objectColor;  
            yield return null;  
        }  
    }  
  
    public void Update()  
    {  
        if (Input.GetKeyDown(KeyCode.Q) && fadeoutin == true)  
        {  
            fadeoutin = false;  
            StartCoroutine(FadeOutObject());  
        }  
        if (Input.GetKeyDown(KeyCode.W) && fadeoutin == false)  
        {  
            fadeoutin = true;  
            StartCoroutine(FadeInObject());  
        }  
    }  
}
```

*In this example 2 Ie numerators are created for fade in and out. Depending on the bool settings it gets activated. The bool Can be changed by using the keys Q (for fade out) and W (for fade in).*

### **Flip sprites left right. Lclip0004.**

<https://youtu.be/GLD5hF6OGBc>

```
///C# Unity library by René Pol///  
///Flip sprites left right (2D)///  
///Tested with unity 2019.3.1.f1///  
  
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
public class Flipsprite : MonoBehaviour  
{  
  
    public bool facingright = true;  
    Vector3 characterScale;  
    float characterScaleX;  
  
    void Start()  
    {  
        characterScale = transform.localScale;  
        characterScaleX = characterScale.x;  
    }  
  
    public void Flips()  
    {  
        if(facingright == true)  
        {  
            characterScale.x = characterScaleX;  
        }  
        if (facingright == false)  
        {  
            characterScale.x = -characterScaleX;  
        }  
    }  
  
    void Update()  
    {  
        Flips();  
        if (Input.GetKeyDown(KeyCode.LeftArrow))  
        {  
            facingright = true;  
        }  
        if (Input.GetKeyDown(KeyCode.RightArrow))  
        {  
            facingright = false;  
        }  
        transform.localScale = characterScale;  
    }  
}
```

*In this example the left and right arrow key set the bool. The flip function is checked continuously and flips the sprite if it needs to face an other direction.*

### **Rotate gameobjects clockwise and counterclockwise. Lclip0005.**

<https://youtu.be/0BC8Ym7VvPc>

```
///C# Unity library by René Pol///  
///Rotate clockwise and counterclockwise (3D-2D)///  
///Tested with unity 2019.3.1.f1///  
  
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
public class Rotate : MonoBehaviour  
{  
    public float speed =50;  
  
    public void Update()  
    {  
        if (Input.GetKey(KeyCode.LeftArrow))  
            transform.Rotate(0f, 0f, speed * Time.deltaTime);  
        if (Input.GetKey(KeyCode.RightArrow))  
            transform.Rotate(0f, 0f, -speed * Time.deltaTime);  
    }  
}
```

*In this example the left and right arrow keys are used to start rotating in the direction needed. It uses the speed variable on the Z axis. (X,Y,Z). The speed can be changed in the Inspectors window. You can rotate using any axis you need.*

### **Top down vehicle/character movement. Lclip0006.**

[https://youtu.be/Mqgh2kGat\\_w](https://youtu.be/Mqgh2kGat_w)

```
///C# Unity library by René Pol///  
///Topdown vehicle/charactermovement (2D)///  
///Tested with unity 2019.3.1.f1///  
  
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
public class Topdownmovement : MonoBehaviour  
{  
    public float rotatespeed = 1;  
    public float movespeed = 4;  
  
    public void Update()  
    {  
        float moveVector = Input.GetAxis("Vertical");  
        float rotateVector = Input.GetAxis("Horizontal");  
        this.transform.Translate(0f, moveVector * movespeed * Time.deltaTime, 0f);  
        this.transform.Rotate(0f, 0f, -rotateVector * (rotatespeed * 80) * Time.deltaTime);  
    }  
}
```

*This is a very basic movement code but it's perfect for atari like games as tanks and others. Use the arrow keys to move your gameobject and set the speed in the inspectors window. This script uses the input settings of Unity but you can use any key or device you want for the movement.*