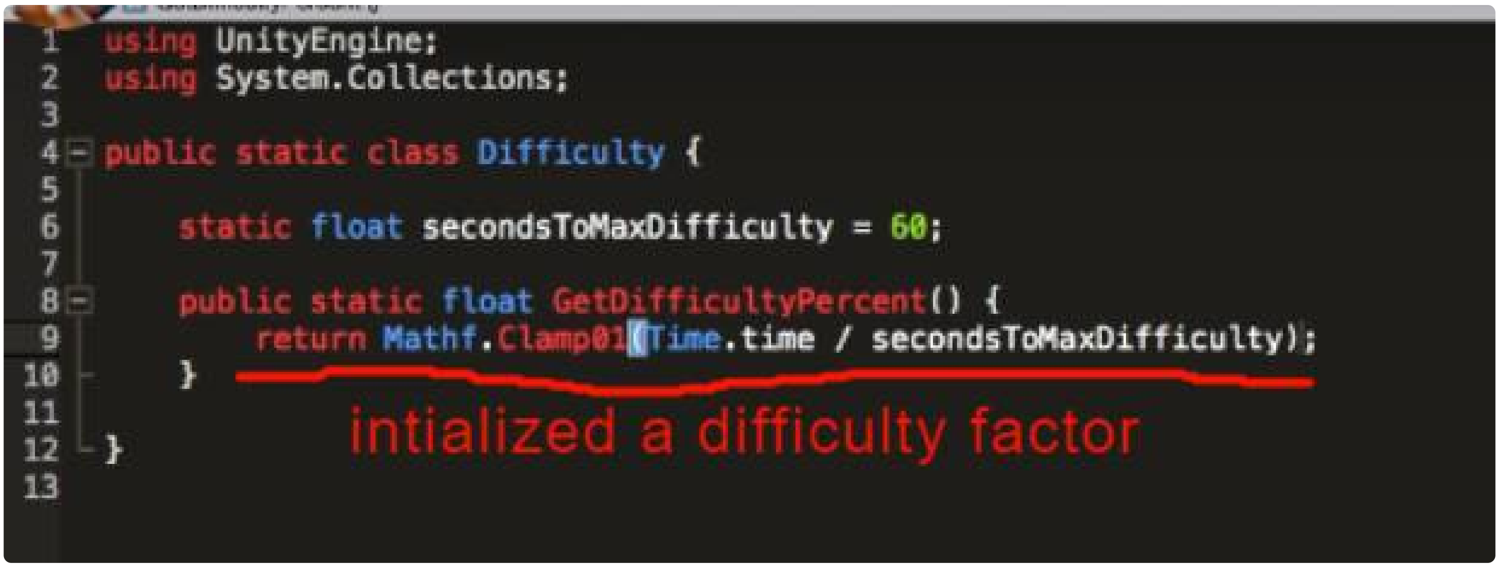
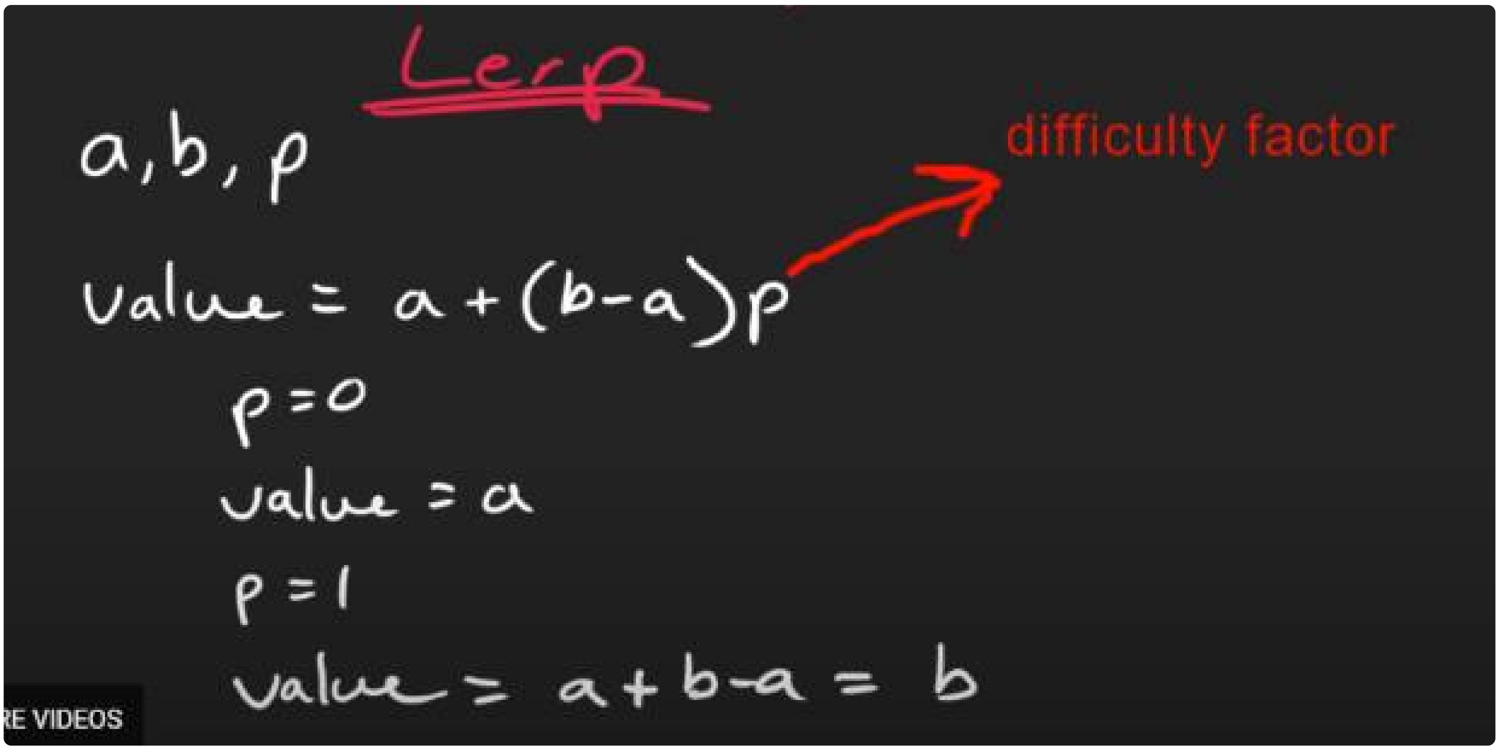


```
using UnityEngine;
using System.Collections;

public static class Difficulty {

    static float secondsToMaxDifficulty = 60;

    public static float GetDifficultyPercent() {
        return Mathf.Clamp01(Time.time / secondsToMaxDifficulty);
    }

}
```

intialized a difficulty factor



Lerp

a, b, p

$$\text{value} = a + (b-a)p$$

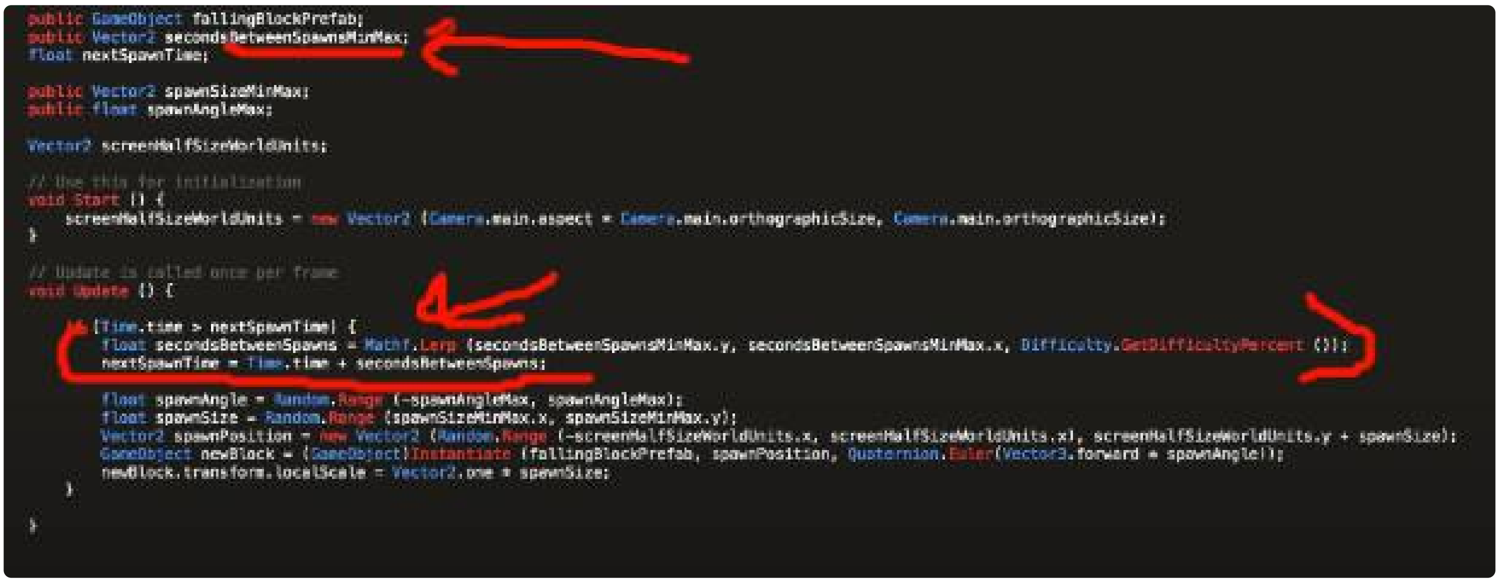
difficulty factor

$p = 0$

$\text{value} = a$

$p = 1$

$\text{value} = a + b - a = b$



```
public GameObject fallingBlockPrefab;
public Vector2 secondsBetweenSpawnsMinMax;
float nextSpawnTime;

public Vector2 spawnSizeMinMax;
public float spawnAngleMax;

Vector2 screenHalfSizeWorldUnits;

// Use this for initialization
void Start () {
    screenHalfSizeWorldUnits = new Vector2 (Camera.main.aspect * Camera.main.orthographicSize, Camera.main.orthographicSize);
}

// Update is called once per frame
void Update () {

    if (Time.time > nextSpawnTime) {
        float secondsBetweenSpawns = Mathf.Lerp (secondsBetweenSpawnsMinMax.y, secondsBetweenSpawnsMinMax.x, Difficulty.GetDifficultyPercent ());
        nextSpawnTime = Time.time + secondsBetweenSpawns;

        float spawnAngle = Random.Range (-spawnAngleMax, spawnAngleMax);
        float spawnSize = Random.Range (spawnSizeMinMax.x, spawnSizeMinMax.y);
        Vector2 spawnPosition = new Vector2 (Random.Range (-screenHalfSizeWorldUnits.x, screenHalfSizeWorldUnits.x), screenHalfSizeWorldUnits.y + spawnSize);
        GameObject newBlock = (GameObject)Instantiate (fallingBlockPrefab, spawnPosition, Quaternion.Euler(Vector3.forward * spawnAngle));
        newBlock.transform.localScale = Vector2.one * spawnSize;
    }

}
```