```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class AttackScript : MonoBehaviour
{
    // Reference to attack's renderer
    Renderer attackRend;

    // Reference to the game controller to access the end game function
    GameController gc;

    // Start and end colors for the Lerp
    // Start is background color, end is red
    public Color startColor;
    public Color endColor;

    // Where on lerp to start and end fade-in
    float startLerp = 0.1f;
    float endLerp = 0.6f;

    // Stores current position on lerp
    float timeAcc;

    // How long to wait after fade-in before suicide
    public float lifetime;

    // Is warning/fade-in happening?
    public bool warningPhase;

    // Length of the warning/fade-in
    public float warningLength;


    void Start()
    {
        // Get game controller script
        gc = GameObject.FindGameObjectWithTag
          ("GameController").GetComponent<GameController>();

        // Get attack's renderer
        attackRend = GetComponent<Renderer>();

        // Set lerp's current position to the start lerp position
        timeAcc = startLerp;

        // Start the warning phase
        warningPhase = true;
```

```
49      }
50
51
52      void Update()
53      {
54          // If it's currently the warning phase...
55          if (warningPhase) {
56
57              // If time accumulation is greater or equal to end lerp (Fade has finished)
58              if (timeAcc >= endLerp) {
59
60                  // Set the color to end lerp, just in case the time accumulation went over
61                  attackRend.material.color = Color.Lerp(startColor, endColor, endLerp);
62
63                  // End the warning phase
64                  warningPhase = false;
65
66                  // Start the Suicide coroutine
67                  StartCoroutine(Suicide());
68
69              } else {
70
71                  // Set color to lerp of time accumulation
72                  attackRend.material.color = Color.Lerp(startColor, endColor, timeAcc);
73
74                  // Time.deltaTime is the time since the last frame. In a perfect world, if the game was running
75                  // at 60 frames per second then after 1 second (ingnoring the division) timeAcc would equal 60
76                  // However, since fade-in ends when timeAcc >= endLerp (0.6), we'll never be able to change the
77                  // fade-in length from 0.6 seconds. To scale Time.deltaTime, it's divided by warningLength.
78                  timeAcc += Time.deltaTime / warningLength;
79              }
80          }
81      }
82
83      IEnumerator Suicide() {
84
85          // Set the color to white, then wait 0.1 seconds. This handles the flash
86          attackRend.material.color = Color.white;
87          yield return new WaitForSeconds(0.1f);
88
```

```
89          // Set the attack's color to full red and move it to z = 0 so the
              player can collide with it
90          attackRend.material.color = endColor;
91          transform.position = new Vector3(transform.position.x,
              transform.position.y, 0);
92
93          // Wait for the duration of lifetime
94          yield return new WaitForSeconds(lifetime);
95
96          // Make the attack the same color as the background, then delete it
97          attackRend.material.color = startColor;
98          Destroy(gameObject);
99      }
100
101     // This is run when the attack collides with something
102     private void OnTriggerEnter(Collider other) {
103
104         // If the object collided with has the tag "Player"...
105         if (other.tag == "Player") {
106
107             // Run the EndGame function in the Game Controller
108             gc.EndGame();
109         }
110     }
111 }
112
```