



Screen Damage Documentation

For support or questions:

visit the Discord server [here](#)
or email me directly at: pathiralgames@gmail.com

GETTING STARTED:

1. Navigate to folder **Screen Damage > Prefab > ScreenDamageCanvas (prefab)**
2. Drag and drop the **ScreenDamageCanvas** prefab to your scene.
3. Select your Camera game object and add component **ScreenDamage**.
4. Drag and drop the **BloodyFrame** child game object from the **ScreenDamageCanvas** (you just put in the scene) into the field "*Bloody Frame*" of Screen Damage inspector.
5. Same thing with "*Blur Effect*", drag and drop the **BlurEffect** child game object from the **ScreenDamageCanvas** into the field "*Blur Effect*" of Screen Damage inspector.
6. Make sure the "*Use Blur Effect*" property is set to **true** and set the "*Blur Duration*" to **0.1** and "*Blur Fade Speed*" to **10**. You can set the numbers however you like this is just the average use.
7. **Your screen damage is now ready to take effect.**
8. Create a new script, let's call it, **HitScreen** and attach it to the camera. We will use this script to hit and decrease health from Screen Damage.
9. Inside the script write:

```
10. using UnityEngine;
11. public class HitScreen : MonoBehaviour
12. {
13.     ScreenDamage script;
14.
15.     void Start()
16.     {
17.         script = GetComponent<ScreenDamage>();
18.     }
19.
20.     void Update()
21.     {
22.         // decrease health
23.         if (Input.GetKeyDown(KeyCode.A)) {
24.             script.CurrentHealth -= 10f;
25.         }
26.
27.
28.
```

```
29.     // increase health
30.     if (Input.GetKeyDown(KeyCode.D)) {
31.         script.CurrentHealth += 10f;
32.     }
33.
34.     //get current health
35.     if (Input.GetKeyDown(KeyCode.Space)) {
36.         Debug.Log(script.CurrentHealth);
37.     }
38. }
39. }
40.
```

Now when you play the game and press A, health is decreased while pressing D increases the health. Pressing spacebar returns the current amount of health.

PUBLIC VARIABLES:

All these variables can be used via code.

currentHealth – returns the current health. You decrease health (hitting the player) by doing `currentHealth -= 10`; This will hit the Player and decrease the Screen Damage by 10. You can also increase health using: `currentHealth += 10`;

maxHealth - the amount of maximum health.

criticalHealth - the amount of health which ought to be critical in order to play the pulse audio.

bloodyFrame - the image component of the bloody frame.

useBlurEffect – if set to true will enable and fade in the radial blur image when hit.

blurImage – the radial blur image that acts as the hit indicator.

blurDuration – the amount of time you want the blur effect to stay after fading in.

blurFadeSpeed – the speed of the fading in/out, the greater the number the faster the fading.

pulseSound - the audio source component of the pulse sound (critical health). This audio will play when health reaches critical health value. Simply set an audio source here and when the health reaches critical value this audio will play. If no audio is set, it'll be skipped.

fadeAudios - turn on/off the audio fader (when in critical health, should certain audio sources fade out so you can only hear the pulse and give a more immersive experience?)

audiosToFade – a list of type AudioSource which contains all the audio sources you want to fade when in critical health.

audiosFadeVolume - the volume to which the world audio sources should fade to when in critical health.

autoHeal - turn on/off auto healing (automatically heal after a number of seconds unhit)

healingSpeed - the speed of the auto heal.

autoHealTime - the amount of time have to be untouched before auto heal kicks in.