Camera Shake Manager for Unity Game Engine.

THE CAMERA SHAKE MANAGAER PROVIDES EXTENSIVELY CONFIGURABLE SHAKE TEMPLATES TO SHAKE THE CAMERA FOR MORE IMPRESSIVE EFFETS.

Last changed: Dec, 2018

Company: metadesc, Helmut Kleber

Author: M. Sc. Dipl. Ing. Helmut Kleber

Contents

1 Introduction

The shake manager consists of three different modules:

- Camera shake
  The camera shake consists of the CameraShakeInfo class, where you can define the camera shake and of the ShakeProcessor, which shakes the camera with the information defined in CameraShakeInfo.
- Camera shake pool
  The pool contains the shake processors to avoid performance spikes in the running game by pre-instantiating the shake processors. The shake pool also limits the instantiation of shakes, to avoid massively overlapping shakes.
- Camera shake tester
  This module allows you to test your shakes. The name of the shake from the pool must be given and optionally a transform (In my example a visible cube, but can be a sumple GameObject instance.). The tester just works in the running game, not in the editor mode.

2 How to use it

There are lot of camera handling solutions. Theoretically the most of them works in the way, that the position and the rotation will be preserved in internal attributes. The position and rotation (sometimes just one of them) will be set after the calculation that the camera script does. The following script part must be added after the script sets the position and rotation. The shake will be an offset at the end, this is the way how shaking can be added to different camera scripts in a modular way.

In the following camera example script the shake script lines are added after the position and rotation have been set. This camera script is an abstract example, this will not move or rotate the camera. This part is also explained in the video.

```
/// <summary> ///
Example for how to add the shake to your camera script.
/// </summary>
public class MyCameraScript : MonoBehaviour {
 Quaternion rot;
 Vector3 pos;
 void Awake() {
  rot = transform.rotation;
  pos = transform.position;
 }

 void Update () {
  // A camera manager sets the position and the rotation.
  transform.rotation = rot;
  transform.position = pos;

  // After this just set the shake position and rotation changes.
  // If your camera script modifies the rotation and position in
  // LateUpdate, then move this to LateUpdate after changing
  // these attributes. This call add a shake offset to the camera
  // position and rotation, that the reason why this works.
  Metadesc.CameraShake.ShakeResult shakeResult =
   Metadesc.CameraShake.ShakeManager.I.UpdateAndGetShakeResult();
  if (shakeResult.DoProcessShake) {
   transform.localPosition += shakeResult.ShakeLocalPos;
   transform.localRotation *= shakeResult.ShakeLocalRot;
  }
 }
}
```

The second step is to add the CameraShake prefab to your scene. Do not add the CameraShake prefab under your camera, eg. add it under the root. Then expand the CameraShake GameObject and select the ShakeManager. Set your camera to the CameraTarget attribute. Thats all, after this you can add some entries to your camera shake tester, that is also under the CameraShake GameObject. Add an entry, or use an entry and set one of the shake names (GameObject names under the ShakeManager GameObject, eg. Deployment, Explosion) and a shake source as a Transform instance (eg. Drag and Drop a GameObject instance from the scene, that will act as the source of the shake). After, run the scene and select the ShakeTester, then you have different buttons for playing, pausing, stopping the shake. In this way you can check that the camera script works fine or not.

The main usage of the shake is normally to call it for an effect (mostly sound FX) from the source code. The ShakeManager provides two methods to generate a shake:

- Non position based, shakes at camera position
  public ShakeProcessor AddShake(string shakeInfoName)
- Position based, shakes until the given distance and normally weaker at higher distance
  public ShakeProcessor AddShake(Vector3 position, string shakeInfoName)

Example for calling in source code: **Metadesc.CameraShake.ShakeManager.I.AddShake(Vector(1, 2, 3), "Deployment");**

For more information, look in the API documentation.