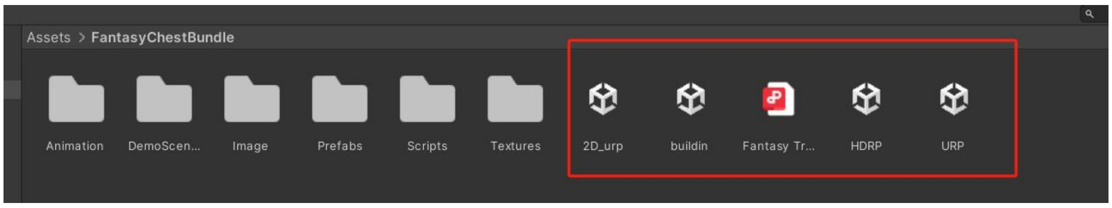
Fantasy Treasure Chest 2D Bundle

1. Installation Guide

1. **Installation Steps:**
   (1) Import package to your Unity project
   (2) Extract secondary package for your current rendering pipeline accordingly
   (3) Notice: The **2D_URP package** is specifically designed for the **2D URP pipeline**. It includes a scene and prefabs for 2D projects.If you selected the default **2D pipeline** when creating your project, please use this **sub-package**.However, if you created your project using a **3D template**, this step is not required—you can still use the **URP sub-package** directly.



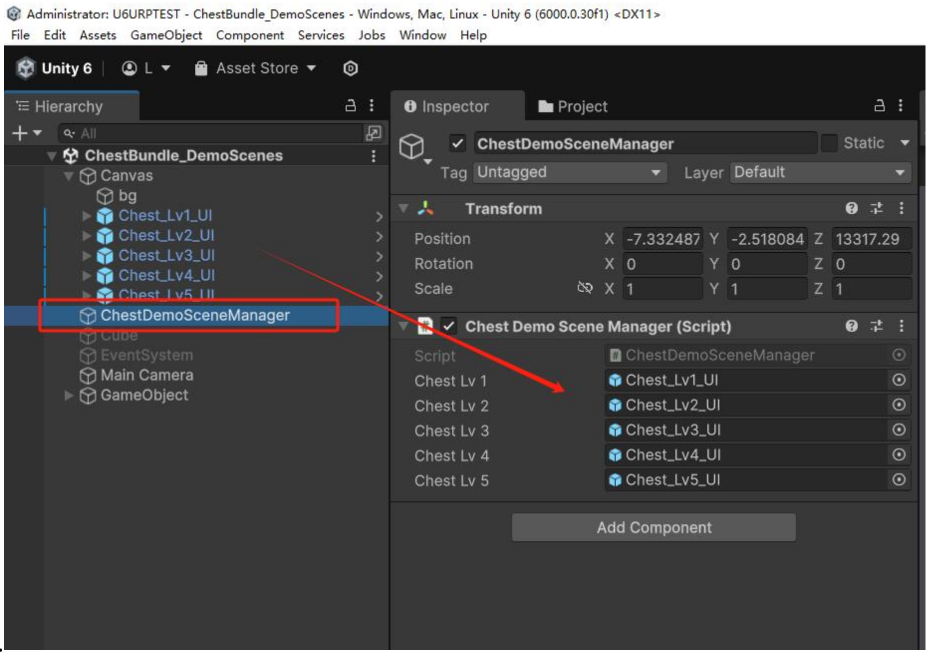
2. **Running the Demo Scene:**
   1. Navigate to the FantasyChestBundle\DemoScenes folder.
   2. For 2d urp project open the scene with a 2D suffix ---"ChestBundle_DemoScenes 2D"
   3. Open the provided Demo Scene.
   4. Press the Play button to run the demo.

3. **How to use:**
   1. Opening and closing the chest must be triggered via code, for example:

```
YourChestGameObject.GetComponent<Chest>().Open();
```

(Go to API section to check out more.)

2. A test script is provided. Open DemoScene , In the Hierarchy, find ChestDemoSceneManager, then enable it. In the Inspector panel, there is a script component named ChestDemoSceneManager. Drag your chest GameObjects into the corresponding object references, then run the scene. The script will wait for 2 seconds and then open all the specified chests. Coding examples are also showed in this script, please also take a look.


Administrator: U6URPTEST - ChestBundle_DemoScenes - Windows, Mac, Linux - Unity 6 (6000.0.30f1) <DX11>

File Edit Assets GameObject Component Services Jobs Window Help
Unity 6
Asset Store
Hierarchy
Inspector
Project
Add Component

```
ChestBundle_DemoScenes
  Canvas
    bg
    Chest_Lv1_UI
    Chest_Lv2_UI
    Chest_Lv3_UI
    Chest_Lv4_UI
    Chest_Lv5_UI
  ChestDemoSceneManager
  Cube
  EventSystem
  Main Camera
  GameObject
```

ChestDemoSceneManager	Static
Tag	Untagged
Layer	Default

Transform

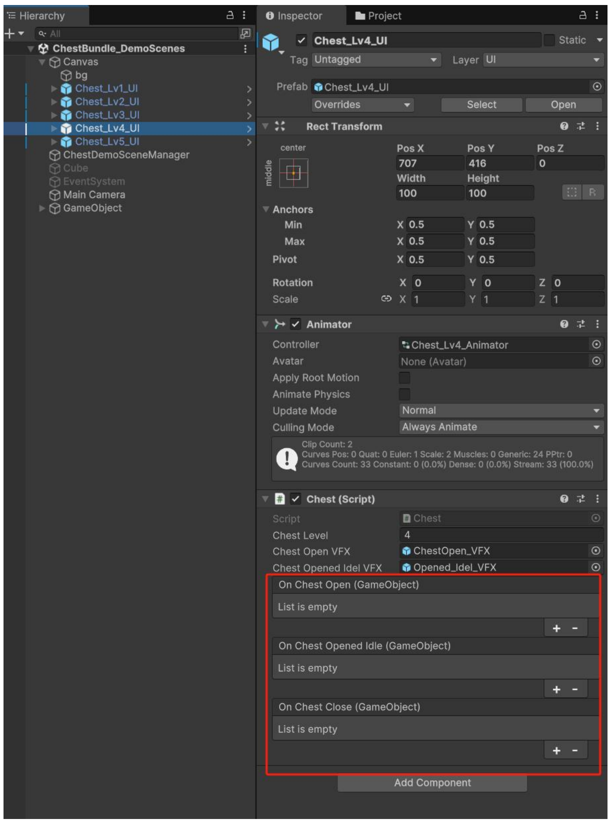
Position	X -7.332487	Y -2.518084	Z 13317.29
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Chest Demo Scene Manager (Script)

Script	ChestDemoSceneManager
Chest Lv 1	Chest_Lv1_UI
Chest Lv 2	Chest_Lv2_UI
Chest Lv 3	Chest_Lv3_UI
Chest Lv 4	Chest_Lv4_UI
Chest Lv 5	Chest_Lv5_UI

.

3. Additionally, If you need to execute your code logic when the chest is opening, closing, or in an idle state, a callback function will be triggered via an animation event during state transitions. You should bind your logic to the UnityEvent<GameObject> on the Chest component, which is handled by OnChestOpen, OnChestOpenedIdle, and OnChestClose, respectively.



```
ChestBundle_DemoScenes
  Canvas
    bg
    Chest_Lv1_UI
    Chest_Lv2_UI
    Chest_Lv3_UI
    Chest_Lv4_UI
    Chest_Lv5_UI
  ChestDemoSceneManager
  Cube
  EventSystem
  Main Camera
  GameObject
```

Chest_Lv4_UI	Static
Tag	Untagged
Layer	UI
Prefab	Chest_Lv4_UI

Rect Transform

center / middle

	Pos X	Pos Y	Pos Z
	707	416	0
	Width	Height	
	100	100	
Anchors			
Min	X 0.5	Y 0.5	
Max	X 0.5	Y 0.5	
Pivot	X 0.5	Y 0.5	
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Animator

Controller	Chest_Lv4_Animator
Avatar	None (Avatar)
Apply Root Motion	☐
Animate Physics	☐
Update Mode	Normal
Culling Mode	Always Animate

```
Clip Count: 2
Curves Pos: 0 Quat: 0 Euler: 1 Scale: 2 Muscles: 0 Generic: 24 PPtr: 0
Curves Count: 33 Constant: 0 (0.0%) Dense: 0 (0.0%) Stream: 33 (100.0%)
```

Chest (Script)

Script	Chest
Chest Level	4
Chest Open VFX	ChestOpen_VFX
Chest Opened Idel VFX	Opened_Idel_VFX

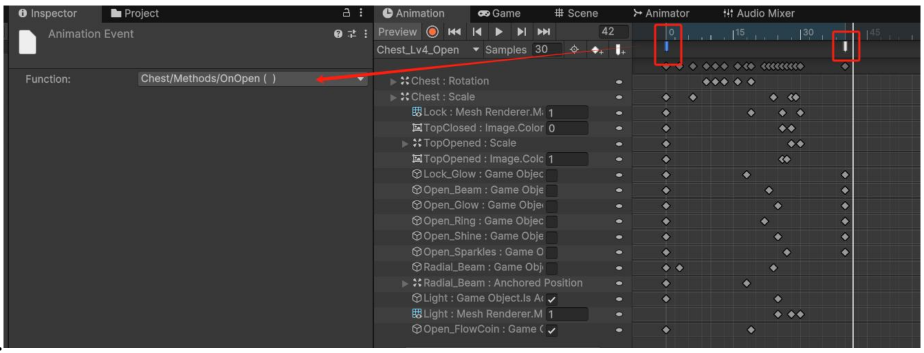
On Chest Open (GameObject)
List is empty

On Chest Opened Idle (GameObject)
List is empty

On Chest Close (GameObject)
List is empty

4. How to customize or rewrite any scripts

1. If you want to modify animation events, locate the white flag in the Animation panel. You can change the associated function, adjust the event's trigger timing, or delete the event if it's not needed



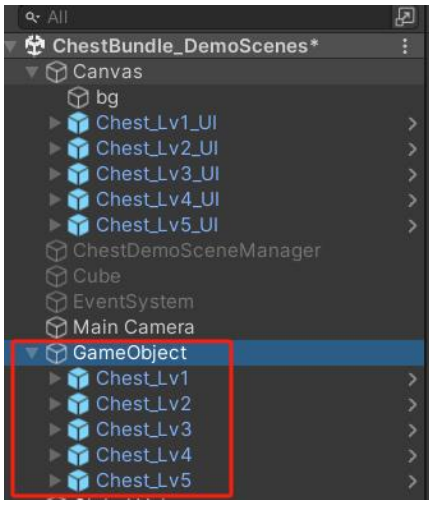
Function: Chest/Methods/OnOpen ( )

.

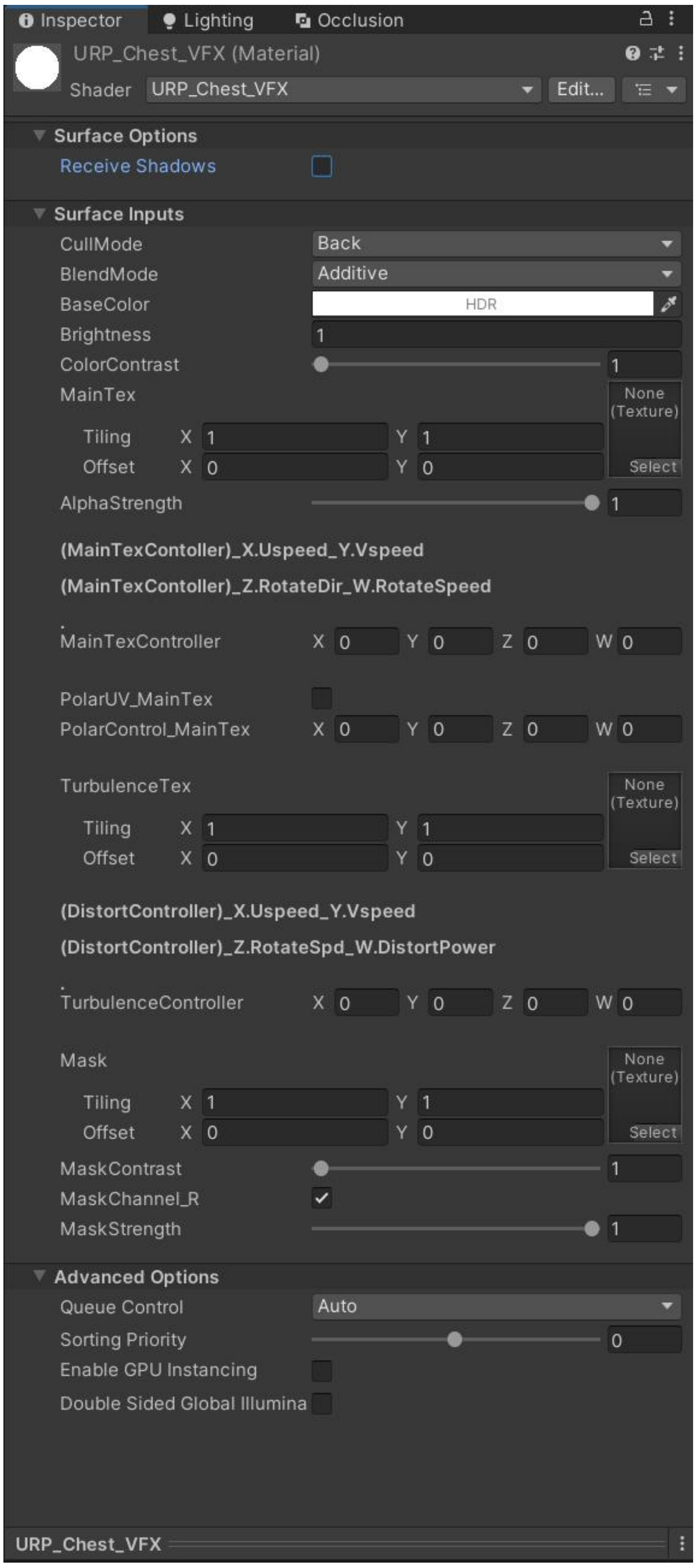
2. All chest-related APIs are contained within the Chest script. Modify this file to customize your chest logic.

Notice:

1. This package provides two types of prefabs for both UGUI and Scene GameObject.
2. Prefabs with the _UI suffix are created using the Image component of UGUI (Path: FantasyChestBundle\Prefabs\UI or FantasyChestBundle\Prefabs\2D_UI). This approach is more convenient and offers better performance when working **solely with UGUI.** Note that in this case, the Canvas in UGUI must be set to **Screen Space - Camera mode**.
3. Another type of prefab can be found in the FantasyChestBundle\Prefabs\Model folder. These prefabs are created using the **Quad Mesh** or **SpriteRenderer** for 2d project . If you prefer to load the prefabs directly into the scene, you should use these prefabs. An Example can be found in the demo scene, Under GameObject in Hierarchy



2. Shader Introduction



URP_Chest_VFX (Material)

Shader URP_Chest_VFX

Surface Options

Receive Shadows ☐

Surface Inputs

CullMode	Back
BlendMode	Additive
BaseColor	HDR
Brightness	1
ColorContrast	1
MainTex	None (Texture)
Tiling	X 1 Y 1
Offset	X 0 Y 0
AlphaStrength	1

(MainTexContoller)_X.Uspeed_Y.Vspeed

(MainTexContoller)_Z.RotateDir_W.RotateSpeed

MainTexController	X 0 Y 0 Z 0 W 0
PolarUV_MainTex	☐
PolarControl_MainTex	X 0 Y 0 Z 0 W 0
TurbulenceTex	None (Texture)
Tiling	X 1 Y 1
Offset	X 0 Y 0

(DistortController)_X.Uspeed_Y.Vspeed

(DistortController)_Z.RotateSpd_W.DistortPower

TurbulenceController	X 0 Y 0 Z 0 W 0
Mask	None (Texture)
Tiling	X 1 Y 1
Offset	X 0 Y 0
MaskContrast	1
MaskChannel_R	✓
MaskStrength	1

Advanced Options

Queue Control	Auto
Sorting Priority	0
Enable GPU Instancing	☐
Double Sided Global Illumina	☐

URP_Chest_VFX

Cull Mode

Off: Renders both front and back sides of the mesh.

Front: Only renders the back side of the mesh, culling the front side.

Back: Only renders the front side of the mesh, culling the back side.

Blend Mode

Additive: Blending mode is SrcAlpha One, creating an additive effect.

AlphaBlend: Blending mode is SrcAlpha OneMinusSrcAlpha

Base Color

A high-dynamic-range (HDR) color that multiplies the MainTex texture.

Brightness

Controls the overall brightness of the output color.

Color Contrast

Adjusts the color contrast by calculating the nth power of the MainTex color.

MainTex

The primary texture used by the shader.

Alpha Strength

Controls the overall transparency of the shader's output.

MainTex Controller

X: Pans the MainTex along the X-axis.

Y: Pans the MainTex along the Y-axis.

Z: Rotates the UV coordinates.

W: Rotates the UV coordinates with a specific speed around the center point (0.5, 0.5).

PolarUV_MainTex

Toggles whether the MainTex samples UV coordinates using a polar axis.

PolarControl_MainTex

X: Radial scale of the polar coordinates.

Y: Length scale of the polar coordinates.

Z: UV panning speed towards or away from the center.

W: Rotates UV coordinates with a specific speed around the center point (0.5, 0.5).

TurbulenceTex

Creates turbulence effects for the MainTex.

TurbulenceTex Controller
X: Pans the turbulence texture along the X-axis.
Y: Pans the turbulence texture along the Y-axis.
Z: Rotates the UV coordinates of the turbulence texture around the center point (0.5, 0.5).
W: Controls the distortion strength.

Mask
Masks the output range of the shader.

Mask Contrast
Adjusts the contrast of the mask by calculating the power of the mask's grayscale values.

MaskChannel_R
Toggles whether the mask samples from the R-channel or the alpha channel.

Mask Strength
Blends between the masked and original output:
A value of 1 fully applies the mask.
A value of 0 means the mask texture has no effect.

○

3. API

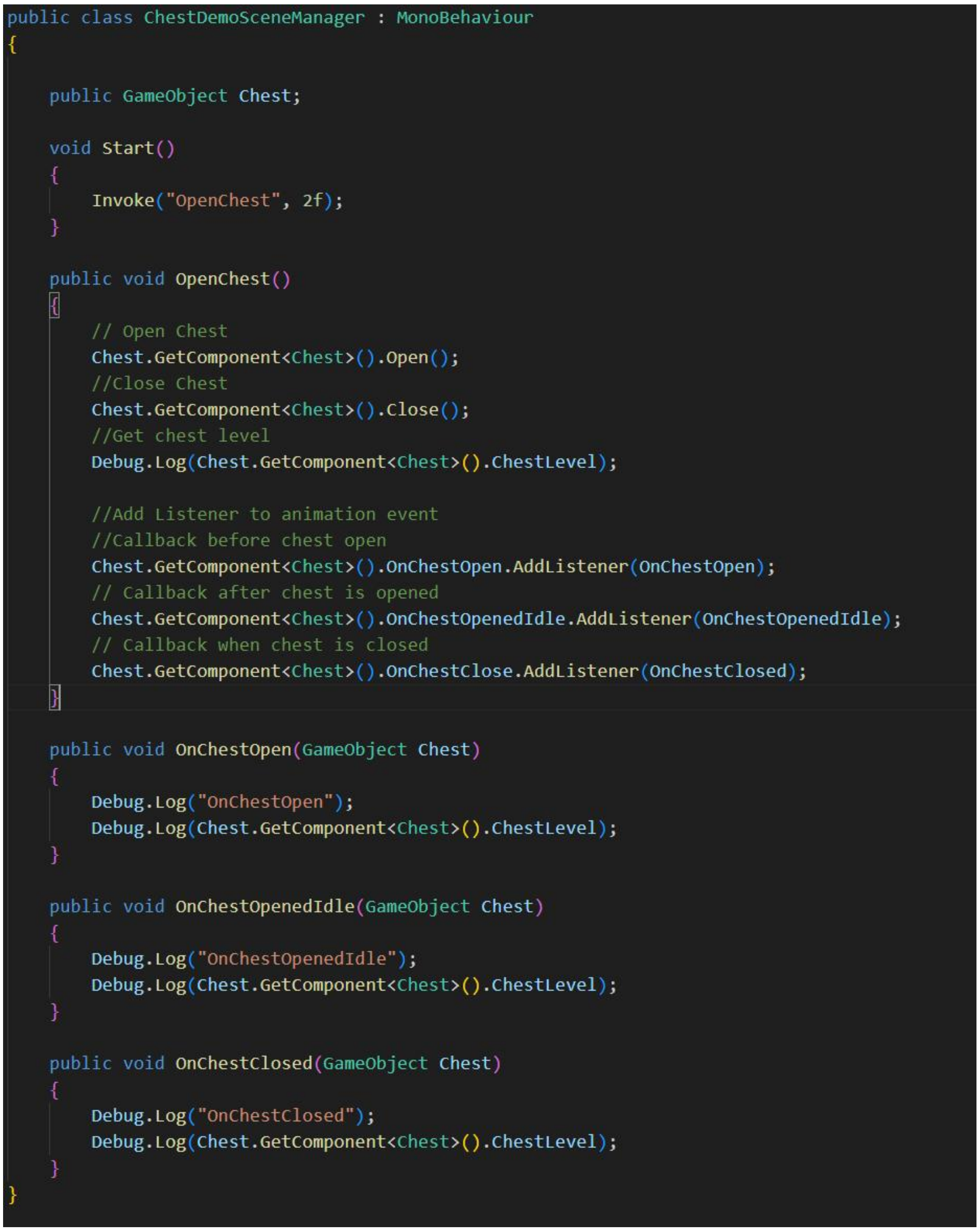
For each chest GameObject prefab, a Chest component is attached. This component provides essential data, animation event callbacks, and methods for controlling animations.

Properties:
```
public int ChestLevel
public Animator ChestAnimator
```

Methods:
```
public void Open()
public void Close()
```

```
Event:
public UnityEvent<GameObject> OnChestOpen;
public UnityEvent<GameObject> OnChestOpenedIdle;
public UnityEvent<GameObject> OnChestClose;
```

Code example



```
public class ChestDemoSceneManager : MonoBehaviour
{

    public GameObject Chest;

    void Start()
    {
        Invoke("OpenChest", 2f);
    }

    public void OpenChest()
    {
        // Open Chest
        Chest.GetComponent<Chest>().Open();
        //Close Chest
        Chest.GetComponent<Chest>().Close();
        //Get chest level
        Debug.Log(Chest.GetComponent<Chest>().ChestLevel);

        //Add Listener to animation event
        //Callback before chest open
        Chest.GetComponent<Chest>().OnChestOpen.AddListener(OnChestOpen);
        // Callback after chest is opened
        Chest.GetComponent<Chest>().OnChestOpenedIdle.AddListener(OnChestOpenedIdle);
        // Callback when chest is closed
        Chest.GetComponent<Chest>().OnChestClose.AddListener(OnChestClosed);
    }

    public void OnChestOpen(GameObject Chest)
    {
        Debug.Log("OnChestOpen");
        Debug.Log(Chest.GetComponent<Chest>().ChestLevel);
    }

    public void OnChestOpenedIdle(GameObject Chest)
    {
        Debug.Log("OnChestOpenedIdle");
        Debug.Log(Chest.GetComponent<Chest>().ChestLevel);
    }

    public void OnChestClosed(GameObject Chest)
    {
        Debug.Log("OnChestClosed");
        Debug.Log(Chest.GetComponent<Chest>().ChestLevel);
    }
}
```
]