```
Blit - Copy.cs - Blit.cs
yEngine;
                                       yEngine;
yEngine.Rendering;
                                       yEngine.Rendering;
yEngine.Rendering.Universal;
                                       yEngine.Rendering.Universal;
3D GalaxyMap;
                                       3D GalaxyMap;
3D Core;
                                       3D Core;
3D Combat;
                                       3D Combat;
Assets.Script
                                       Assets.Script
t Renderer Feature
                                       t Renderer Feature
ed on the Blit from the UniversalRende ed on the Blit from the UniversalRende
ps://github.com/Unity-Technologies/Uni ps://github.com/Unity-Technologies/Uni
ended to allow for :
                                      ended to allow for :
pecific access to selecting a source a pecific access to selecting a source a
utomatic switching to using AfterPost utomatic switching to using AfterPost
etting a InverseView matrix (cameraTo etting a InverseView matrix (cameraTo
  e.g. Reconstruct world pos from dept e.g. Reconstruct world pos from dept
URP v10) Enabling generation of DepthN URP v10) Enabling generation of DepthN
  This will only include shaders who he This will only include shaders who he
  (workaround for Unlit Shaders / Grap (workaround for Unlit Shaders / Grap
                                       anilux
anilux
eAssetMenu(menuName = "Feature/Blit")] eAssetMenu(menuName = "Feature/Blit")]
 class Blit : ScriptableRendererFeatur class Blit : ScriptableRendererFeatur
blic class BlitPass : ScriptableRender blic class BlitPass : ScriptableRender
  public Material blitMaterial = null;
                                         public Material blitMaterial = null;
  public FilterMode filterMode { get;
                                         public FilterMode filterMode { get;
  private BlitSettings settings;
                                         private BlitSettings settings;
  private RTHandle source { get; set;
                                         private RTHandle source { get; set;
  private RTHandle destination { get;
                                         private RTHandle destination { get;
  RTHandle m TemporaryColorTexture;
                                         RTHandle m TemporaryColorTexture;
  RTHandle m DestinationTexture;
                                         RTHandle m DestinationTexture;
  string m ProfilerTag;
                                         string m ProfilerTag;
  public BlitPass(RenderPassEvent rend
                                         public BlitPass(RenderPassEvent rend
      this.renderPassEvent = renderPas
                                             this.renderPassEvent = renderPas
      this.settings = settings;
                                             this.settings = settings;
```

```
Blit - Copy.cs - Blit.cs
     blitMaterial = settings.blitMate
                                             blitMaterial = settings.blitMate
      m ProfilerTag = tag;
                                             m ProfilerTag = tag;
      //m TemporaryColorTexture.Init("
                                             //m TemporaryColorTexture.Init("
      //if (settings.dstType == Target
                                             //if (settings.dstType == Target
      //{
                                             //{
      // m DestinationTexture.Init(se
                                             // m DestinationTexture.Init(se
      //}
                                             //}
 public void Setup(RTHandle source, R
                                         public void Setup(RTHandle source, R
      this.source = source;
                                             this.source = source;
      this.destination = destination;
                                             this.destination = destination;
                                       2020 1 OR NEWER
2020 1 OR NEWER
                                             if (settings.requireDepthNormals
      if (settings.requireDepthNormals
          ConfigureInput(ScriptableRen
                                                 ConfigureInput(ScriptableRen
 public override void Execute (Scripta
                                         public override void Execute(Scripta
      CommandBuffer cmd = CommandBuffe
                                             CommandBuffer cmd = CommandBuffe
      RenderTextureDescriptor opaqueDe
                                             RenderTextureDescriptor opaqueDe
      opaqueDesc.depthBufferBits = 0;
                                             opaqueDesc.depthBufferBits = 0;
      if (settings.setInverseViewMatri
                                             if (settings.setInverseViewMatri
          Shader.SetGlobalMatrix(" Inv
                                                 Shader.SetGlobalMatrix(" Inv
      if (settings.dstType == Target.T
                                             if (settings.dstType == Target.T
          if (settings.overrideGraphic
                                                 if (settings.overrideGraphic
              opaqueDesc.graphicsForma
                                                     opaqueDesc.graphicsForma
          cmd.GetTemporaryRT(0, 1, 1,
                                                 cmd.GetTemporaryRT(m Destina
      //Debug.Log($"src = {source},
                                             //Debug.Log($"src = {source},
      // Can't read and write to same
                                             // Can't read and write to same
      if (source == destination || (se
                                             if (source == destination || (se
          cmd.GetTemporaryRT(0, 1, 1,
                                                 cmd.GetTemporaryRT(m Tempora
          Blit(cmd, source, destinatio
                                                 Blit(cmd, source, destination
          Blit(cmd, source, destinatio
                                                 //Blit(cmd, m TemporaryColor
```

```
else
                                             else
          Blit(cmd, source, destinatio
                                                 Blit (cmd, source, destinatio
      context.ExecuteCommandBuffer(cmd
                                             context.ExecuteCommandBuffer(cmd
      CommandBufferPool.Release(cmd);
                                             CommandBufferPool.Release(cmd);
                                         public override void FrameCleanup (Co
  public override void FrameCleanup(Co)
      if (settings.dstType == Target.T
                                             if (settings.dstType == Target.T
          cmd.ReleaseTemporaryRT(0);
                                                 cmd.ReleaseTemporaryRT(m Des
      if (source == destination || (se
                                             if (source == destination || (se
          cmd.ReleaseTemporaryRT(0);
                                                 cmd.ReleaseTemporaryRT(m Tem
ystem.Serializable]
                                       ystem.Serializable]
blic class BlitSettings
                                       blic class BlitSettings
  public RenderPassEvent Event = Rende
                                         public RenderPassEvent Event = Rende
  public Material blitMaterial = null;
                                         public Material blitMaterial = null;
  public int blitMaterialPassIndex = 0
                                         public int blitMaterialPassIndex = 0
  public bool setInverseViewMatrix = f
                                         public bool setInverseViewMatrix = f
  public bool requireDepthNormals = fa
                                         public bool requireDepthNormals = fa
  public Target srcType = Target.Camer
                                         public Target srcType = Target.Camer
                                         public string srcTextureId = " Camer
  public string srcTextureId = " Camer
                                         public RenderTexture srcTextureObjec
  public RenderTexture srcTextureObjec
  public Target dstType = Target.Camer
                                         public Target dstType = Target.Camer
                                         public string dstTextureId = " BlitP
  public string dstTextureId = " BlitP
  public RenderTexture dstTextureObjec
                                         public RenderTexture dstTextureObjec
  public bool overrideGraphicsFormat =
                                         public bool overrideGraphicsFormat =
  public UnityEngine.Experimental.Rend
                                         public UnityEngine.Experimental.Rend
blic enum Target
                                       blic enum Target
  CameraColor,
                                         CameraColor,
  TextureID,
                                         TextureID,
  RenderTextureObject
                                         RenderTextureObject
```

Blit - Copy.cs - Blit.cs

```
blic BlitSettings settings = new();
blic BlitSettings settings = new BlitS
blic BlitPass blitPass;
                                   blic BlitPass blitPass;
ivate RTHandle srcIdentifier, dstIdent ivate RTHandle srcIdentifier, dstIdent
blic override void Create()
                                   blic override void Create()
 var passIndex = settings.blitMateria var passIndex = settings.blitMateria
  settings.blitMaterialPassIndex = Mat
                                     settings.blitMaterialPassIndex = Mat
 blitPass = new BlitPass(settings.Eve
                                     blitPass = new BlitPass (settings.Eve
  if (settings.Event == RenderPassEven
                                     if (settings.Event == RenderPassEven
     Debug.LogWarning("Note that the
                                         Debug.LogWarning("Note that the
  if (settings.graphicsFormat == Unity
                                     if (settings.graphicsFormat == Unity
     settings.graphicsFormat = System
                                         settings.graphicsFormat = System
  //UpdateSrcIdentifier();
                                     //UpdateSrcIdentifier();
  //UpdateDstIdentifier();
                                     //UpdateDstIdentifier();
srcIdentifier = UpdateIdentifier(set
                                     srcIdentifier = UpdateIdentifier(set
private void UpdateDstIdentifier()
                                   private void UpdateDstIdentifier()
  dstIdentifier = UpdateIdentifier(set
                                     dstIdentifier = UpdateIdentifier(set
if (type == Target.RenderTextureObje
                                     if (type == Target.RenderTextureObje
     return obj;
                                         return obj;
  else if (type == Target.TextureID)
                                     else if (type == Target.TextureID)
     m RTHandle.Init(s);
                                         //RenderTargetHandle m RTHandle
                                         //m RTHandle.Init(s);
     return new RTHandle (this, type);
     //return s;
                                         //return m RTHandle.Identifier()
                                         return s;
  return new RTHandle ();
```

blic override void AddRenderPasses (Scr blic override void AddRenderPasses (Scr

```
if (settings.blitMaterial == null)
                                       if (settings.blitMaterial == null)
    Debug.LogWarningFormat("Missing
                                           Debug.LogWarningFormat("Missing
    return;
                                           return;
if (settings.Event == RenderPassEven
                                       if (settings.Event == RenderPassEven
else if (settings.Event == RenderPas
                                       else if (settings.Event == RenderPas
    // If event is AfterRendering, a
                                           // If event is AfterRendering, a
    if (settings.srcType == Target.C
                                           if (settings.srcType == Target.C
        settings.srcType = Target.Te
                                               settings.srcType = Target.Te
        settings.srcTextureId = " Af
                                               settings.srcTextureId = " Af
        //UpdateSrcIdentifier();
                                               //UpdateSrcIdentifier();
    if (settings.dstType == Target.C
                                           if (settings.dstType == Target.C
        settings.dstType = Target.Te
                                               settings.dstType = Target.Te
        settings.dstTextureId = " Af
                                               settings.dstTextureId = " Af
        //UpdateDstIdentifier();
                                               //UpdateDstIdentifier();
else
                                       else
    // If src/dst is using AfterPos
                                           // If src/dst is using AfterPos
                                           if (settings.srcType == Target.T
    if (settings.srcType == Target.T
        settings.srcType = Target.Ca
                                               settings.srcType = Target.Ca
        settings.srcTextureId = "";
                                               settings.srcTextureId = "";
        //UpdateSrcIdentifier();
                                               //UpdateSrcIdentifier();
    if (settings.dstType == Target.T
                                           if (settings.dstType == Target.T
        settings.dstType = Target.Ca
                                               settings.dstType = Target.Ca
        settings.dstTextureId = "";
                                               settings.dstTextureId = "";
        //UpdateDstIdentifier();
                                               //UpdateDstIdentifier();
var src = (settings.srcType == Targe var src = (settings.srcType == Targe
var dest = (settings.dstType == Targ
                                      var dest = (settings.dstType == Targ
```

Blit - Copy.cs - Blit.cs

```
blitPass.Setup(src, dest);
renderer.EnqueuePass(blitPass);
blitPass.Setup(src, dest);
renderer.EnqueuePass(blitPass);
```