DOCUMENTATION

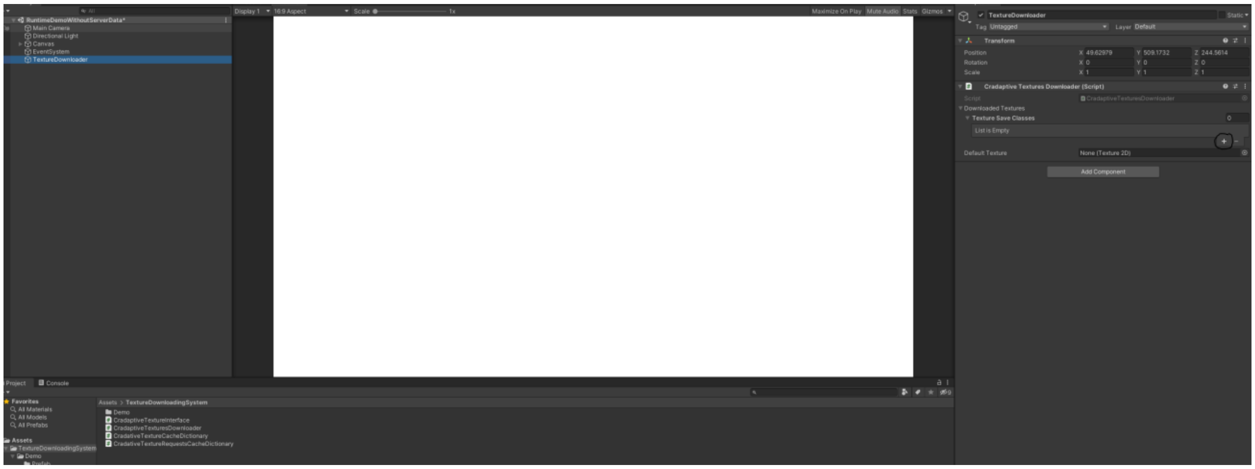
MAKING LOCAL TEXTURES AVAILABLE

METHOD 1:

This system can access local textures if you put all textures in the folder “Resources/CradaptiveTextures”. You can have multiple folders of this.

METHOD 2:

Create a new gameobject, attach the “`CradaptiveTexturesDownloader`” component to it.



TextureDownloader
Tag Untagged
Layer Default
Position X 49.62979 Y 509.1732 Z 244.5614
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Script CradaptiveTexturesDownloader
Texture Save Classes 0
List is Empty
Default Texture None (Texture 2D)

Click on the plus button and add your local textures you which to have available.

Make sure to assign the sprite as well as the url or name you which to use as a key to access this sprite.

GETTING TEXTURES FROM SERVER DATA

If you are using this tool with data that comes from a server. For example, you expect a json response with a url link. Just implement the interface in your base data class.

Example :

```
[System.Serializable]
public class CradaptiveSimpleData : ICradaptiveTextureOwner
{
    public string name;
    /// <summary>
    /// Url for image or local name in the Downloaded textures dictionary (if
  you  have the image locally assigned)
    /// </summary>
    public string url { get => serialisedUrl; set => serialisedUrl = value; }

    public string serialisedUrl;
    /// <summary>
    /// Callback for when image is available
```

```
        /// </summary>
        public Action<Sprite> OnTextureAvailable { get; set; }
    }
```

Now in your prefab, you can easily ask for a texture by passing that data to the texture download system. By doing the following:

You only need to set the callback which expects an action with a sprite variable. This would be called as soon as a response returns. Note that if your texture is local. This should be almost instant.

```
public void Init(CradaptiveSimpleData cradaptiveSimpleData)
{
    cradaptiveSimpleData.OnTextureAvailable = (spr) =>
    {
        if(image)
        image.sprite = spr;
    };
    CradaptiveTexturesDownloader.QueueForDownload(cradaptiveSimpleData);
}
```

GETTING TEXTURES FROM LOCAL DATA

If you intend to use the local textures, and have probably assigned the textures to the system or placed in the resources folder. You can easily access the textures anywhere by initializing a struct.

```
public void Init()
{
    TextureDownloadRequest textureDownloadRequest = new TextureDownloadRequest();
    textureDownloadRequest.url = url;
    textureDownloadRequest.OnTextureAvailable = (spr) =>
    {
        image.sprite = spr;
    };
    CradaptiveTexturesDownloader.QueueForDownload(textureDownloadRequest);
}
```

You set the callback as usual. As well as the url or local name you have assigned. This way you don’t need to create a new class for every texture request.